

AlphaServer 1200

DIGITAL Ultimate Workstation 533

Service Manual

Order Number: EK-AS120-SV. A01

This manual is for anyone who services these systems. It includes troubleshooting information, configuration rules, and instructions for removal and replacement of field-replaceable units.

Digital Equipment Corporation
Maynard, Massachusetts

First Printing, January 1998

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software, if any, described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license. No responsibility is assumed for the use or reliability of software or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1998 by Digital Equipment Corporation. All rights reserved.

The following are trademarks of Digital Equipment Corporation: AlphaServer, OpenVMS, StorageWorks, VAX, and the DIGITAL logo.

The following are third-party trademarks: Lifestyle 28.8 DATA/FAX Modem is a trademark of Motorola, Inc. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Ltd. U.S. Robotics and Sportster are registered trademarks of U.S. Robotics. Windows NT is a trademark of Microsoft, Inc. All other trademarks and registered trademarks are the property of their respective holders.

FCC Notice: The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense will be required to take whatever measures are required to correct the interference.

Shielded Cables: If shielded cables have been supplied or specified, they must be used on the system in order to maintain international regulatory compliance.

Warning! This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

Achtung! Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

Avertissement! Cet appareil est un appareil de Classe A. Dans un environnement résidentiel, cet appareil peut provoquer des brouillages radioélectriques. Dans ce cas, il peut être demandé à l'utilisateur de prendre les mesures appropriées.

Contents

Preface	xi
---------------	----

Chapter 1 Overview

1.1	System Enclosure	1-2
1.2	Operator Control Panel and Drives	1-4
1.3	System Consoles	1-6
1.4	System Architecture	1-8
1.5	CPU Types.....	1-10
1.6	Memory	1-12
1.7	Memory Addressing	1-14
1.8	System Motherboard	1-16
1.8.1	System Bus (Backplane).....	1-18
1.8.2	System Bus to PCI Bus Bridge.....	1-20
1.8.3	PCI I/O Subsystem	1-22
1.8.4	Remote Control Logic	1-24
1.8.5	Power Control Logic	1-26
1.9	Power Circuit and Cover Interlock.....	1-28
1.10	Power Supply.....	1-30
1.11	Power Up/Down Sequence	1-32
1.12	Maintenance Bus (I ² C Bus).....	1-34
1.13	StorageWorks.....	1-36

Chapter 2 Power-Up

2.1	Control Panel	2-2
2.2	Power-Up Sequence	2-4
2.3	SROM Power-Up Test Flow.....	2-8
2.4	SROM Errors Reported	2-11
2.5	XSROM Power-Up Test Flow	2-12
2.6	XSROM Errors Reported	2-15
2.7	Console Power-Up Tests	2-16
2.8	Console Device Determination	2-18
2.9	Console Power-Up Display.....	2-20
2.10	Fail-Safe Loader.....	2-24

Chapter 3 Troubleshooting

3.1	Troubleshooting with LEDs.....	3-2
3.2	Troubleshooting Power Problems	3-4
3.3	Running Diagnostics — Test Command	3-6
3.4	Releasing Secure Mode	3-7
3.5	Testing an Entire System	3-8
3.5.1	Testing Memory	3-10
3.5.2	Testing PCI	3-12
3.6	Other Useful Console Commands	3-14

Chapter 4 Error Logs

4.1	Using Error Logs.....	4-2
4.1.1	Hard Errors	4-4
4.1.2	Soft Errors.....	4-4
4.1.3	Error Log Events	4-5
4.2	Using DECEvent	4-6
4.2.1	Translating Event Files	4-7
4.2.2	Filtering Events	4-8
4.2.3	Selecting Alternative Reports.....	4-10
4.3	Error Log Examples and Analysis.....	4-11
4.3.1	MCHK 670 CPU-Detected Failure.....	4-11
4.3.2	MCHK 670 CPU and IOD-Detected Failure.....	4-16
4.3.3	MCHK 670 Read Dirty CPU-Detected Failure	4-21
4.3.4	MCHK 660 IOD-Detected Failure (System Bus Error).....	4-27
4.3.5	MCHK 660 IOD-Detected Failure (PCI Error)	4-32
4.3.6	MCHK 630 Correctable CPU Error.....	4-41
4.3.7	MCHK 620 Correctable Error	4-44
4.4	Troubleshooting IOD-Detected Errors	4-47
4.4.1	System Bus ECC Error	4-48
4.4.2	System Bus Nonexistent Address Error	4-49
4.4.3	System Bus Address Parity Error	4-50
4.4.4	PIO Buffer Overflow Error (PIO_OVFL).....	4-51
4.4.5	Page Table Entry Invalid Error	4-52
4.4.6	PCI Master Abort	4-52
4.4.7	PCI System Error	4-52
4.4.8	PCI Parity Error.....	4-52
4.4.9	Broken Memory	4-53
4.4.10	Command Codes	4-55
4.4.11	Node IDs.....	4-56
4.5	Double Error Halts and Machine Checks While in PAL Mode	4-57
4.5.1	PALcode Overview	4-57
4.5.2	Double Error Halt.....	4-58

4.5.3	Machine Checks While in PAL.....	4-58
-------	----------------------------------	------

Chapter 5 Error Registers

5.1	External Interface Status Register - EL_STAT	5-2
5.2	External Interface Address Register - EI_ADDR	5-6
5.3	MC Error Information Register 0 (MC_ERR0 - Offset = 800)	5-8
5.4	MC Error Information Register 1 (MC_ERR1 - Offset = 840)	5-9
5.5	CAP Error Register (CAP_ERR - Offset = 880)	5-11
5.6	PCI Error Status Register 1 (PCI_ERR1 - Offset = 1040)	5-14

Chapter 6 Removal and Replacement

6.1	System Safety	6-1
6.2	FRU List	6-2
6.3	System Exposure.....	6-6
6.4	CPU Removal and Replacement	6-8
6.5	CPU Fan Removal and Replacement	6-10
6.6	Memory Riser Card Removal and Replacement	6-12
6.7	DIMM Removal and Replacement.....	6-14
6.8	System Motherboard Removal and Replacement.....	6-16
6.9	PCI/EISA Option Removal and Replacement.....	6-18
6.10	Power Supply Removal and Replacement	6-20
6.11	Power Harness Removal and Replacement.....	6-22
6.12	System Fan Removal and Replacement.....	6-24
6.13	Cover Interlock Removal and Replacement.....	6-26
6.14	Operator Control Panel Removal and Replacement	6-28
6.15	CD-ROM Removal and Replacement	6-30
6.16	Floppy Removal and Replacement.....	6-32
6.17	SCSI Disk Removal and Replacement	6-34
6.18	StorageWorks Backplane Removal and Replacement.....	6-36
6.19	StorageWorks Ultra SCSI Bus Extender Removal and Replacement.....	6-38

Appendix A Running Utilities

A.1	Running Utilities from a Graphics Monitor	A-2
A.2	Running Utilities from a Serial Terminal	A-3
A.3	Running ECU.....	A-4
A.4	Running RAID Standalone Configuration Utility	A-5
A.5	Updating Firmware with LFU.....	A-6
A.5.1	Updating Firmware from the CD-ROM	A-8
A.5.2	Updating Firmware from the Floppy Disk — Creating the Diskettes	A-12
A.5.3	Updating Firmware from the Floppy Disk — Performing the Update	A-14
A.5.4	Updating Firmware from a Network Device.....	A-18
A.5.5	LFU Commands.....	A-22

A.6	Updating Firmware from AlphaBIOS	A-25
A.7	Upgrading AlphaBIOS	A-26

Appendix B Halts, Console Commands, and Environment Variables

B.1	Halt Button Functions.....	B-2
B.2	Using the Halt Button	B-3
B.3	Halt Assertion	B-4
B.1	Summary of SRM Console Commands	B-6
B.1.1	Summary of SRM Environment Variables	B-8
B.2	Recording Environment Variables	B-10

Appendix C Running Utilities

C.1	RCM Overview	C-2
C.2	First-Time Setup	C-3
C.2.1	Configuring the Modem	C-4
C.2.2	Dialing In and Invoking RCM	C-5
C.2.3	Using RCM Locally	C-6
C.3	RCM Commands.....	C-7
C.4	Dial-Out Alerts.....	C-16
C.5	Using the RCM Switchpack.....	C-19
C.6	Troubleshooting Guide	C-23
C.7	Modem Dialog Details	C-26

Index

Examples

2-1	SRAM Errors Reported at Power-Up	2-11
2-2	XSROM Errors Reported at Power-Up	2-15
3-1	Test Command Syntax	3-6
3-2	Releasing/Reestablishing Secure Mode.....	3-7
3-3	Sample Test Command.....	3-8
3-4	Sample Test Memory Command.....	3-10
3-5	Sample Test Command for PCI	3-12
3-6	Show Power.....	3-14
3-7	Show Memory.....	3-14
3-8	Show FRU	3-15
4-1	MCHK 670	4-12
4-2	MCHK 670 CPU and IOD-Detected Failure	4-17
4-3	MCHK 670 Read Dirty Failure	4-22
4-4	MCHK 660 IOD-Detected Failure (System Bus Error).....	4-28
4-5	MCHK 660 IOD-Detected Failure (PCI Error).....	4-33

4-6	MCHK 630 Correctable CPU Error	4-42
4-7	MCHK 620 Correctable Error.....	4-45
4-8	INFO 3 Command.....	4-59
4-9	INFO 5 Command.....	4-61
4-10	INFO 8 Command.....	4-63
A-1	Starting LFU from the SRM Console	A-5
A-2	Booting LFU from the CD-ROM.....	A-6
A-3	Updating Firmware from the Internal CD-ROM.....	A-7
A-4	Creating Update Diskettes on an OpenVMS System.....	A-12
A-5	Updating Firmware from the Internal Floppy Disk	A-13
A-6	Selecting AS1200FW to Update Firmware from the Internal Floppy	A-16
A-7	Updating Firmware from a Network Device.....	A-17
C-1	Sample Remote Dial-In Dialog	C-5
C-2	Invoking and Leaving RCM Locally	C-6
C-3	Configuring the Modem for Dial-Out Alerts	C-16
C-4	Typical RCM Dial-Out Command	C-17

Figures

1-1	System Enclosure	1-2
1-2	Cover Interlock Circuit.....	1-3
1-3	Control Panel Assembly	1-4
1-4	Architecture Diagram	1-8
1-5	CPU Module Placement	1-10
1-6	Memory Placement	1-12
1-7	How Memory Addressing Is Calculated.....	1-14
1-8	System Motherboard	1-16
1-9	System Bus Block Diagram	1-18
1-10	System Bus to PCI Bus Bridge Block Diagram	1-20
1-11	PCI Block Diagram	1-22
1-12	Remote Control Logic	1-24
1-13	Power Control Logic	1-26
1-14	Power Circuit Diagram.....	1-28
1-15	Back of Power Supply and Location	1-30
1-16	Power Up/Down Sequence Flowchart.....	1-32
1-17	I ² C Bus Block Diagram	1-34
1-18	StorageWorks Drive Location.....	1-36
2-1	Control Panel and LCD Display.....	2-2
2-2	Power-Up Flow	2-4
2-3	Contents of FEPROMs	2-5
2-4	Console Code Critical Path (1200 Block Diagram).....	2-6
2-5	SRROM Power-Up Test Flow.....	2-8
2-6	XSROM Power-Up Flowchart.....	2-12
2-7	Console Device Determination Flowchart.....	2-18

3-1	System Motherboard LEDs.....	3-2
4-1	Error Detector Placement	4-2
6-1	System FRU Locations.....	6-2
6-2	Exposing the System	6-6
6-3	Removing CPU Module	6-8
6-4	Removing CPU Fan	6-10
6-5	Removing Memory Riser Card	6-12
6-6	Removing A DIMM from a Memory Riser Card.....	6-14
6-7	Removing System Motherboard.....	6-16
6-8	Removing PCI/EISA Option.....	6-18
6-9	Removing Power Supply	6-20
6-10	Removing Power Harness.....	6-22
6-11	Removing System Fan.....	6-24
6-12	Removing Cover Interlocks	6-26
6-13	Removing OCP	6-28
6-14	Removing CD_ROM.....	6-30
6-15	Removing Floppy.....	6-32
6-16	Removing StorageWorks Disk.....	6-34
6-17	Removing StorageWorks Backplane.....	6-36
6-18	Removing StorageWorks Ultra SCSI Bus Extender.....	6-38
A-1	Running a Utility from a Graphics Monitor.....	A-2
A-2	Starting LFU from the AlphaBIOS Console	A-6
A-3	AlphaBIOS Setup Screen	A-25
A-1	System Partition Not Defined	A-29
C-1	RCM Connections	C-3
C-2	Location of RCM Switchpack on System Board.....	C-19
C-3	RCM Switches (Factory Settings).....	C-20

Tables

1-1	PCI Motherboard Slot Numbering	1-23
2-1	Control Panel Display.....	2-3
2-2	SROM Tests.....	2-10
2-3	XSROM Tests.....	2-13
2-4	Memory Tests	2-14
2-5	IOD Tests.....	2-16
2-6	PCI Motherboard Tests.....	2-17
4-1	Types of Error Log Events.....	4-5
4-2	DECEvent Report Formats.....	4-10
4-3	CAP Error Register Data Pattern.....	4-47
4-4	System Bus ECC Error Data Pattern	4-48
4-5	System Bus Nonexistent Address Error Troubleshooting.....	4-49
4-6	Address Parity Error Troubleshooting.....	4-50
4-7	Cause of PIO_OVFL Error	4-51

4-8	ECC Syndrome Bits Table.....	4-54
4-9	Decoding Commands	4-55
4-10	Node IDs.....	4-56
5-1	External Interface Status Register	5-4
5-2	Loading and Locking Rules for External Interface Registers	5-7
5-3	MC Error Information Register 0.....	5-8
5-4	MC Error Information Register 1	5-10
5-5	CAP Error Register	5-12
5-6	PCI Error Status Register 1	5-14
6-1	Field-Replaceable Unit Part Numbers	6-3
A-1	AlphaBIOS Option Key Mapping.....	A-3
A-2	File Locations for Creating Update Diskettes on a PC	A-12
A-3	LFU Command Summary.....	A-22
B-1	Results of Pushing the Halt Button	B-2
B-2	Summary of SRM Console Commands.....	B-6
B-3	Environment Variable Summary	B-8
C-1	RCM Command Summary	C-7
C-2	RCM Status Command Fields.....	C-15
C-3	Elements of the Dial-Out String.....	C-18
C-4	RCM Troubleshooting.....	C-23

Preface

Intended Audience

This manual is written for the customer service engineer.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections for efficient online and printed reference. Each topic begins with an abstract, followed by an illustration or example, and ends with descriptive text.

This manual has six chapters and three appendixes, as follows:

- **Chapter 1, System Overview**, introduces the DIGITAL AlphaServer 1200 and the DIGITAL Ultimate Workstation 533 systems. It describes each system component.
- **Chapter 2, Power-Up**, provides information on how to interpret the power-up display on the operator control panel, the console screen, and system LEDs. It also describes how hardware diagnostics execute when the system is initialized.
- **Chapter 3, Troubleshooting**, describes troubleshooting during power-up and booting, as well as the **test** command.
- **Chapter 4, Error Logs**, explains how to interpret error logs and how to use DECEvent.
- **Chapter 5, Error Registers**, describes the error registers used to hold error information.
- **Chapter 6, Removal and Replacement**, describes removal and replacement procedures for field-replaceable units (FRUs).
- **Appendix A, Running Utilities**, explains how to run utilities such as the EISA Configuration Utility and RAID Standalone Configuration Utility.
- **Appendix B, Halts, Console Commands, and Environment Variables**, summarizes the commands used to examine and alter the system configuration.
- **Appendix C, Operating the System Remotely**, describes how to use the Remote Console Manager (RCM) to monitor and control the system remotely.

Documentation Titles

Table 1 lists books in the documentation set for both systems.

Table 1 System Documentation

Title	Order Number
User and Installation Documentation Kit	QZ-011AA-GW
<i>AlphaServer 1200 User's Guide</i>	EK-AS120-UG
<i>AlphaServer 1200 Basic Installation</i>	EK-AS120-IG
User and Installation Documentation Kit	QZ-013AA-GW
<i>DIGITAL Ultimate Workstation 533 User's Guide</i>	EK-UW120-UG
<i>DIGITAL Ultimate Workstation 533 Basic Installation</i>	EK-UW120-IG
Service Information	
<i>AlphaServer 1200 /DIGITAL Ultimate Workstation 533 Service Manual</i>	EK-AS120-SV

Information on the Internet

Using a Web browser you can access the AlphaServer InfoCenter at:

<http://www.digital.com/info/alphaserver/products.html>

Access the latest system firmware either with a Web browser or via FTP as follows:

<ftp://ftp.digital.com/pub/Digital/Alpha/firmware/>

Interim firmware released since the last firmware CD is located at:

<ftp://ftp.digital.com/pub/Digital/Alpha/firmware/interim/>

Chapter 1

System Overview

The DIGITAL AlphaServer 1200 and DIGITAL Ultimate Workstation 533 systems are made from the same base system unit. The base unit consists of up to two CPUs, up to 2 Gbytes of memory, 6 I/O slots, and up to 7 SCSI storage devices. Both systems are enclosed in pedestals. AlphaServer 1200 systems can be mounted in a standard 19" rack.

AlphaServer 1200 systems support OpenVMS, DIGITAL UNIX, and Windows NT. Ultimate Workstation 533 systems support Windows NT and graphics.

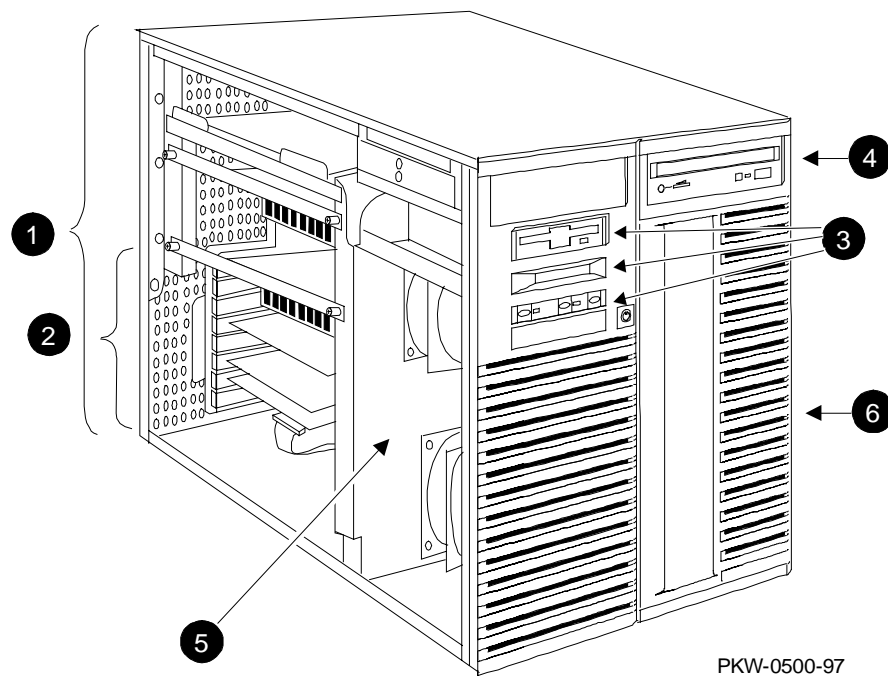
Topics in this chapter include the following:

- System Enclosure
- Operator Control Panel and Drives
- System Consoles
- System Architecture
- CPU Types
- Memory
- Memory Addressing
- System Motherboard
- System Bus Backplane
- System Bus to PCI Bus Bridge
- PCI I/O Subsystem
- Remote Control Logic
- Power Control Logic
- Power Circuit and Cover Interlock
- Power Supply
- Power Up/Down Sequence
- Maintenance Bus (I²C Bus)
- StorageWorks Drives

1.1 System Enclosure

The system has up to two CPU modules and up to 2 Gbytes of memory. A single fast wide or fast wide Ultra SCSI StorageWorks shelf provides storage.

Figure 1-1 System Enclosure



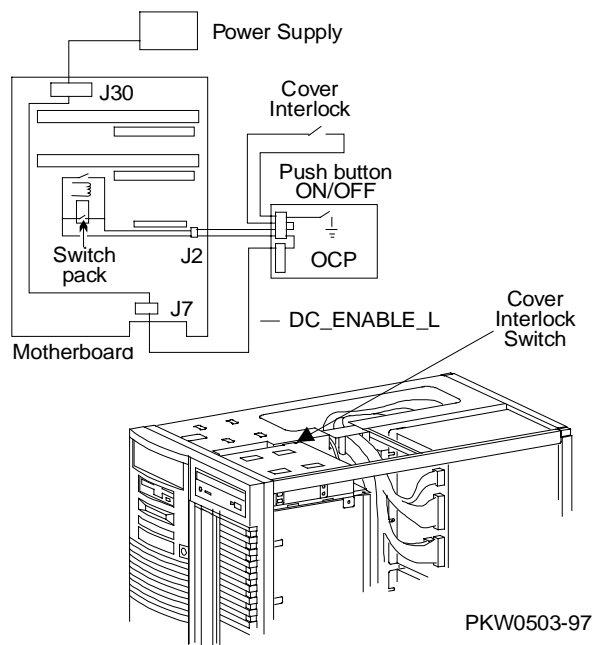
The numbered callouts in Figure 1-1 refer to the system components.

- ❶ System card cage, which holds the system motherboard and the CPU, memory, and system I/O.
- ❷ PCI/EISA section of the system card cage.
- ❸ Operator control panel assembly, which includes the control panel, the LCD display, and the floppy drive.
- ❹ CD-ROM drive.
- ❺ Cooling section containing two fans.
- ❻ StorageWorks shelf.

Cover Interlock

The system has a single cover interlock switch tripped by the top cover.

Figure 1-2 Cover Interlock Circuit



NOTE: The cover interlock must be engaged to enable power-up.

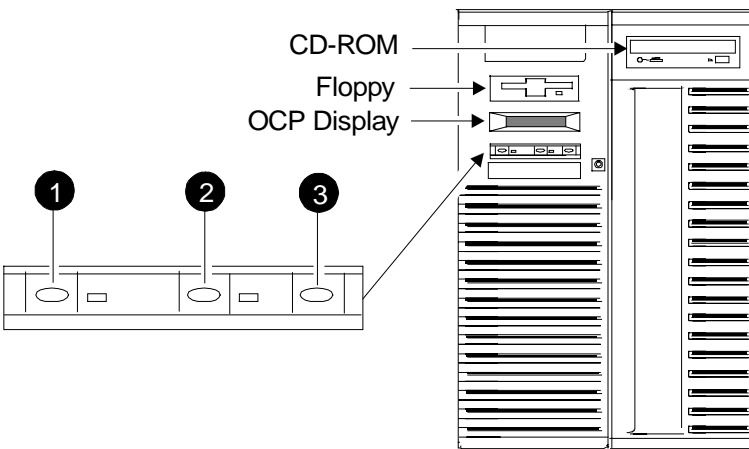
To override the cover interlock, use a suitable object to close the interlock circuit.

Disk damage will result if the system is run with the top cover off.

1.2 Operator Control Panel and Drives

The control panel includes the On/Off, Halt, and Reset buttons and an LCD display.

Figure 1-3 Control Panel Assembly



PKW-0501-97

OCP display. The OCP display is a 16-character LCD that indicates status during power-up and self-test. While the operating system is running, the LCD displays the system type. Its controller is on the XBUS.

CD-ROM. The CD-ROM drive is used to load software, firmware, and updates. Its controller is on PCI1 on the PCI backplane on the system motherboard.

Floppy disk. The floppy drive is used to load software and firmware updates. The floppy controller is on the XBUS on the PCI backplane on the system motherboard.

- ❶ **On/Off button.** Powers the system on or off. When the LED to the right of the button is lit, the power is on. The On/Off button is connected to the power supplies through the system interlock and the RCM logic.
- ❷ **Reset button.** Initializes the system.
- ❸ **Halt button.** When the halt button is pressed, different results are manifest depending upon the state of the machine.

The major function of the Halt button is to stop whatever the machine is doing and return the system to the SRM console.

To get to the SRM console, for systems running OpenVMS or DIGITAL UNIX press the Halt button.

To get to the SRM console, for systems running Windows NT press the Halt button and then press the Reset button. (Pressing the Halt button when the system is running Windows NT causes a “halt assertion” flag to be set in the firmware. When Reset is pressed the console reads the “halt assertion” flag and ignores environment variables that would cause the system to boot.)

Function of the Halt button is complex because it depends upon the state of the machine when the button is pressed. See Section B.1 for a full discussion of the Halt button.

1.3 System Consoles

There are two console programs: the SRM console and the AlphaBIOS console.

SRM Console Prompt

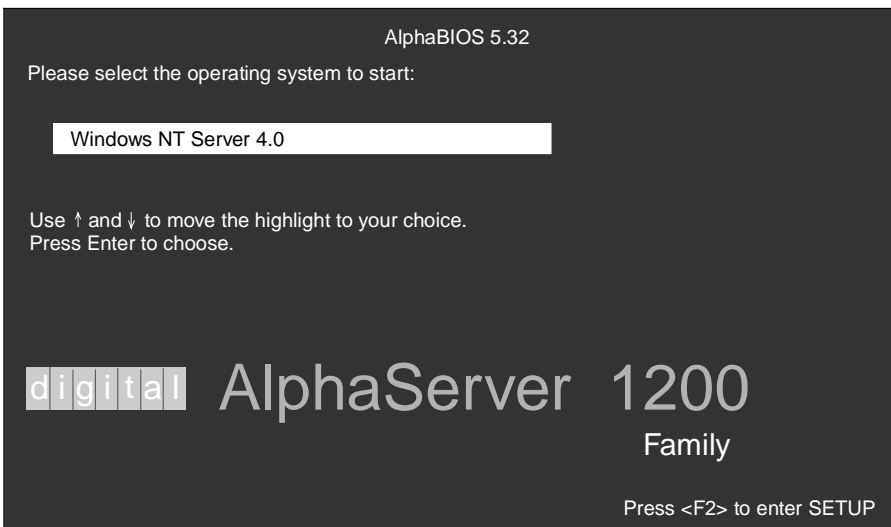
On systems running the DIGITAL UNIX or OpenVMS operating system, the following console prompt is displayed after system startup messages are displayed, or whenever the SRM console is invoked:

P00>>>

NOTE: The console prompt displays only after the entire power-up sequence is complete. This can take up to several minutes if the memory is very large.

AlphaBIOS Boot Menu

On systems running the Windows NT operating system, the Boot menu is displayed when the AlphaBIOS console is invoked:



PKW0560-97

SRM Console

The SRM console is a command-line interface that is used to boot the DIGITAL UNIX and OpenVMS operating systems. It also provides support for examining and modifying the system state and configuring and testing the system. The SRM console can be run from a serial terminal or a graphics monitor.

AlphaBIOS Console

The AlphaBIOS console is a menu-based interface that supports the Microsoft Windows NT operating system. AlphaBIOS is used to set up operating system selections, boot Windows NT, and display information about the system configuration. The EISA Configuration Utility and the RAID Standalone Configuration Utility are run from the AlphaBIOS console. AlphaBIOS runs on either a serial or graphics terminal. Windows NT requires a graphics monitor.

Environment Variables

Environment variables are software parameters that define, among other things, the system configuration. They are used to pass information to different pieces of software running in the system at various times. The **os_type** environment variable, which can be set to VMS, UNIX, or NT, determines which of the two consoles is used. The SRM console is always brought into memory, but AlphaBIOS is loaded if **os_type** is set to **NT** and the Halt LED is not lit.

Refer to Appendix B of this guide for a list of the environment variables used to configure a system.

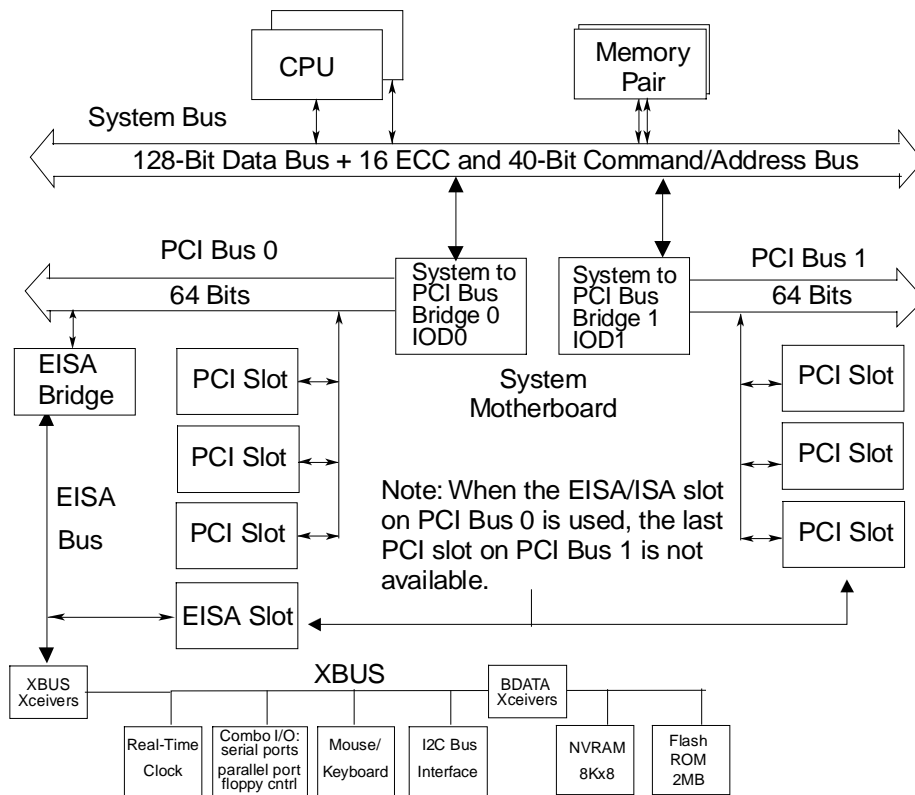
Refer to your system *User's Guide* for information on setting environment variables.

Most environment variables are stored in the NVRAM that is placed in a socket on the system motherboard. Even though the NVRAM can be removed and replaced on a new system motherboard, it is recommended that you keep a record of the environment variables for each system that you service. Some environment variable settings are lost when a module is swapped and must be restored after the new module is installed. Refer to Appendix B for a convenient worksheet for recording environment variable settings.

1.4 System Architecture

Alpha microprocessor chips are used in these systems. The CPU, memory, and the I/O modules are connected to the system motherboard.

Figure 1-4 Architecture Diagram



PKW0502-97

Both systems use the Alpha chip for the CPU. The CPU, memory, and I/O devices connect to the system motherboard. On the system motherboard is:

- The system bus
- Two system bus to PCI bus chip sets that bridge two PCI buses to the system bus
- Two 64-bit PCI buses with three PCI options slots each
- One EISA/ISA bus bridged to one of the PCIs (If an EISA/ISA option is used, one PCI slot cannot be used)
- One CD-ROM controller built in to the other PCI
- One EISA/ISA to XBUS bridge to the built-in XBUS options

A fully configured system can have two CPUs, eight DIMM memory pairs, and a total of six I/O options. The I/O options can be all PCI options or a combination of PCI options and a single EISA/ISA option.

The system bus has a 144-bit data bus, protected by 16 bits of ECC, and a 40-bit command/address bus, protected by parity. The bus speed is set to 66.6 MHz. The 40-bit address bus can create one terabyte of addresses (that's a million million). The bus connects CPUs, memory, and the system bus to PCI bus bridge(s).

There is a cache external to the CPU chip on CPU modules. The Alpha chip has an 8-Kbyte instruction cache (I-cache), an 8-Kbyte write-through data cache (D-cache), and a 96-Kbyte, write-back secondary data cache (S-cache). The cache system is write-back. The system supports up to two CPUs.

Memory on these systems is constructed of DIMM memory pairs placed onto two memory modules called riser cards. The riser cards are placed into the two memory slots on the system motherboard. One member of a DIMM pair is placed onto one riser card, and the other member is placed onto another riser card. Each riser card drives half of the system bus, along with the associated ECC bits. Memory pairs consist of two synchronous DIMMs of the same size and are placed into the same slot on each riser card.

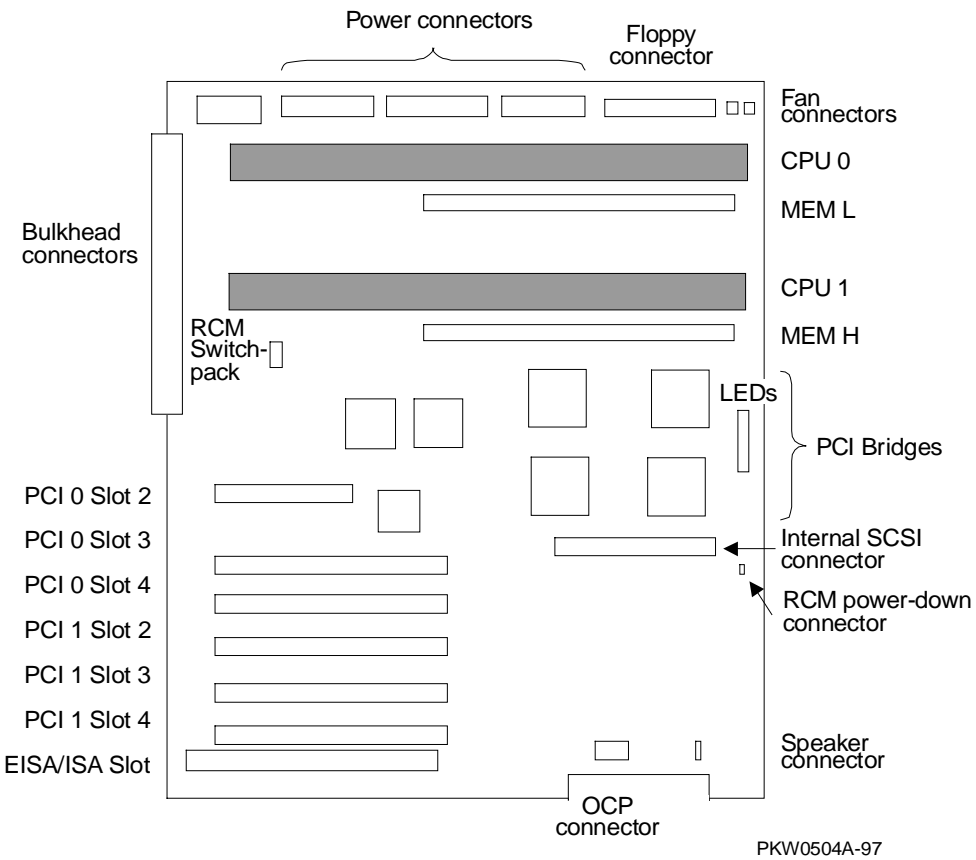
The system bus to PCI bus bridge chip set translates system bus commands and data addressed to I/O space to PCI commands and data. It also translates PCI bus commands and data addressed to system memory or CPUs to system bus commands and data. The PCI bus is a 64-bit wide bus used for I/O.

Logic and sensors on the system motherboard monitor power status and the system environment (temperature and fan speeds).

1.5 CPU Types

There are several CPU variants differentiated by CPU speeds.

Figure 1-5 CPU Module Placement



Alpha Chip Composition

The Alpha chip is made using state-of-the-art chip technology, has a transistor count of 9.3 million, consumes 50 watts of power, and is air cooled (a fan is on the chip). The default cache system is write-back and when the module has an external cache, it is write-back. The Alpha chip used in these systems is the 21164.

Chip Description

Unit	Description
Instruction Execution	8-Kbyte cache, 4-way issue 4-way execution; 2 integer units, 1 floating-point adder, 1 floating-point multiplier
Memory	Merge logic, 8-Kbyte write-through first-level data cache, 96-Kbyte write-back second-level data cache, bus interface unit

CPU Variants

Module Variant	Clock Frequency	Onboard Cache	Color
B3007-AA	400 MHz	4 Mbytes	Orange
B3007-CA	533 MHz	4 Mbytes	Violet

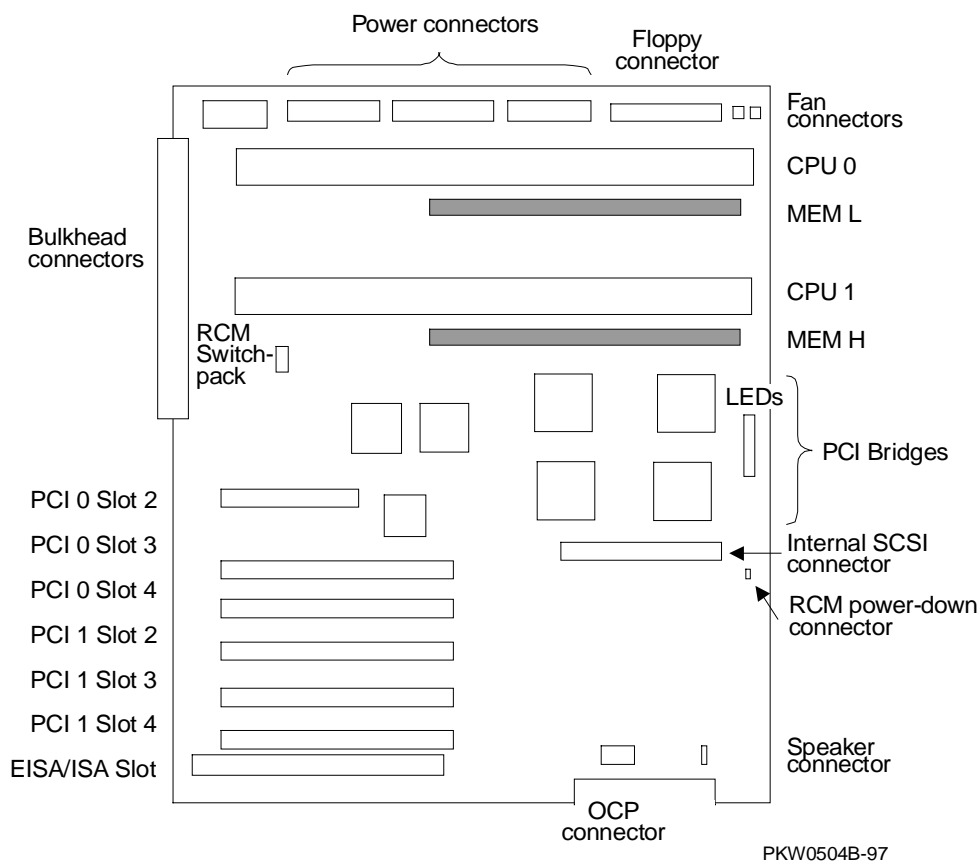
CPU Configuration Rules

- The first CPU must be in CPU slot 0 to provide the system clock.
- The second CPU should be installed in CPU slot 1.
- Both CPUs must have the same Alpha chip clock speed. The system bus may hang without an error message if the oscillators clocking the CPUs are different.

1.6 Memory

Memory consists of two riser cards and up to eight pairs of DIMMs. Each riser card receives one of the two DIMMs in the DIMM pair. There are two DIMM variants: a 32-Mbyte version and a 128-Mbyte version.

Figure 1-6 Memory Placement



Memory Variants

Memory consists of two riser cards supporting eight DIMM pairs. There are two DIMM variants: a 32-Mbyte version and a 128-Mbyte version. Maximum memory using 32-Mbyte DIMMs is 128 Mbytes and the maximum memory using 128-Mbyte DIMMs is 2 Gbytes. All memory is synchronous.

Option	Size	Module	Type	DRAM	
				Number	Size
MS300-BA	64 MB	54-25084-DA 20-47405-D3	Synch.	18	4M x 72 = 32MB
MS300-DA	256 MB	54-25092-DA 20-45619-D3	Synch.	18	16M x 72 = 128MB

Memory Operation

Each DIMM in the pair provides half the data, or 64 bits plus 8 ECC bits, of the octaword (16 byte) transferred on the system bus. DIMMs are placed in slots on the riser cards, and the riser cards are placed in the slots designated MEM L and MEM H on the system motherboard.

NOTE: Memory in slot MEM L does not drive the lower 8 bytes, and memory in slot MEM H does not drive the higher 8 bytes of the 16-byte transfer. Some bits originating from MEM L are high order bits, and some bits originating from MEM H are low order bits.

Memory drives the system bus in bursts. Upon each memory fetch, data is transferred in 4 consecutive cycles transferring 64 bytes.

Memory Configuration Rules

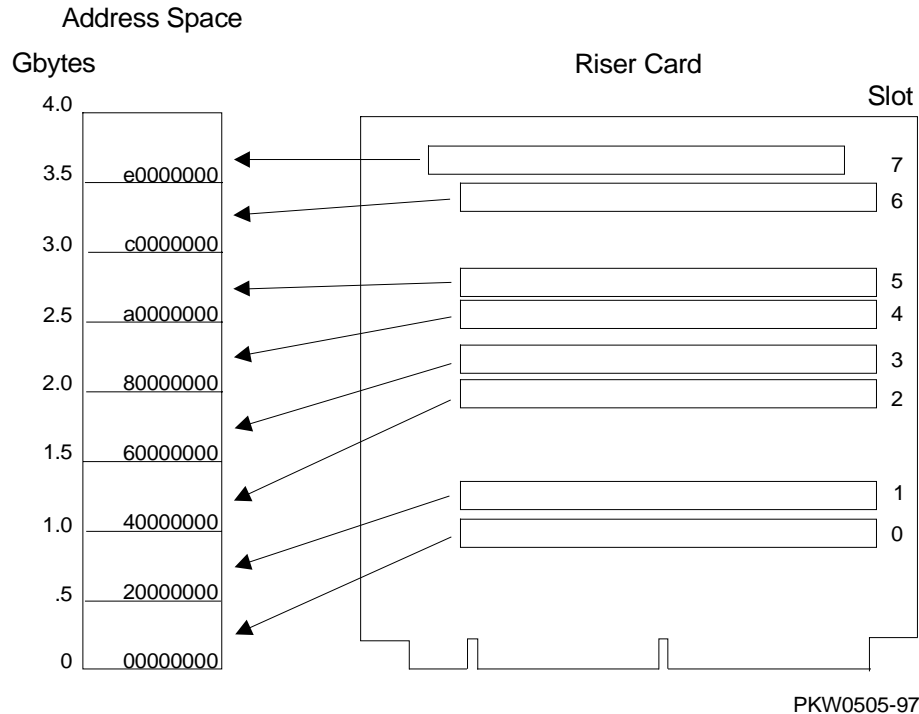
In a system, memories of different sizes are permitted, but:

- DIMMs are installed and used in pairs. Both DIMMs in a memory pair must be of the same size.
- Each riser card receives one DIMM of the DIMM pair.
- The largest DIMM pair must be in riser card slot 0.
- Other memory pairs must be the same size or smaller than the first memory pair.
- Memory pairs must be installed in consecutive slots.
- Memory configurations that have a 64-Mbyte pair in riser card slot 0 are limited to two DIMM pairs or 128 Mbytes for the system. (The reason for this restriction is that the bit map describing memory holes can grow larger than physical memory.)

1.7 Memory Addressing

Memory addressing in these systems is fixed regardless of the size of the DIMMs. The address of a DIMM pair is fixed according to the slot in which the pair is placed. The starting address of each pair in each slot on the riser card starts on a 512-Mbyte boundary.

Figure 1-7 How Memory Addressing Is Calculated



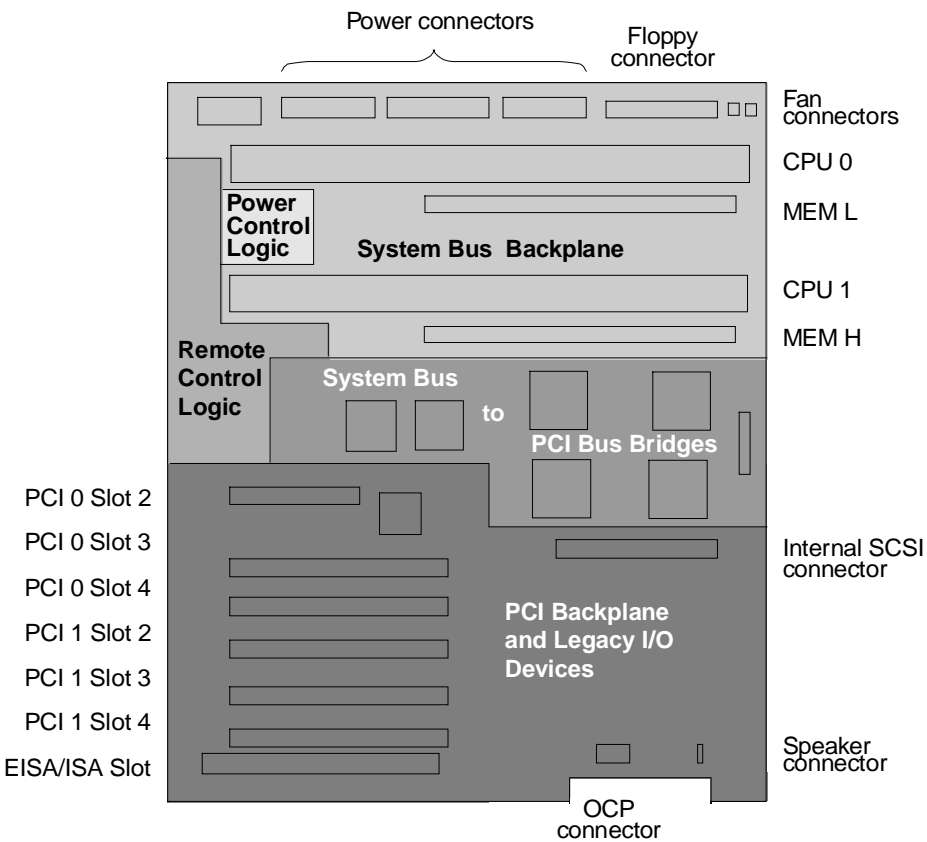
The rules for addressing memory are as follows:

1. A memory pair consists of two DIMMs of the same size.
2. Memory pairs in riser cards may be of different sizes.
3. The memory pair in slot 0 must be the largest of all memory pairs. Other memory pairs may be as large but none may be larger.
4. The physical starting address of each memory pair is N times 512 Mbytes (200 0000) where N is the slot number on the riser card.
5. Memory addresses are contiguous within each memory pair.
6. If memory pairs do not completely fill the 512-Mbyte space provided, memory “holes” occur in the physical address space.
7. Software creates contiguous virtual memory even though physical memory may not be contiguous.

1.8 System Motherboard

The system motherboard contains five major logic sections performing five major system functions.

Figure 1-8 System Motherboard



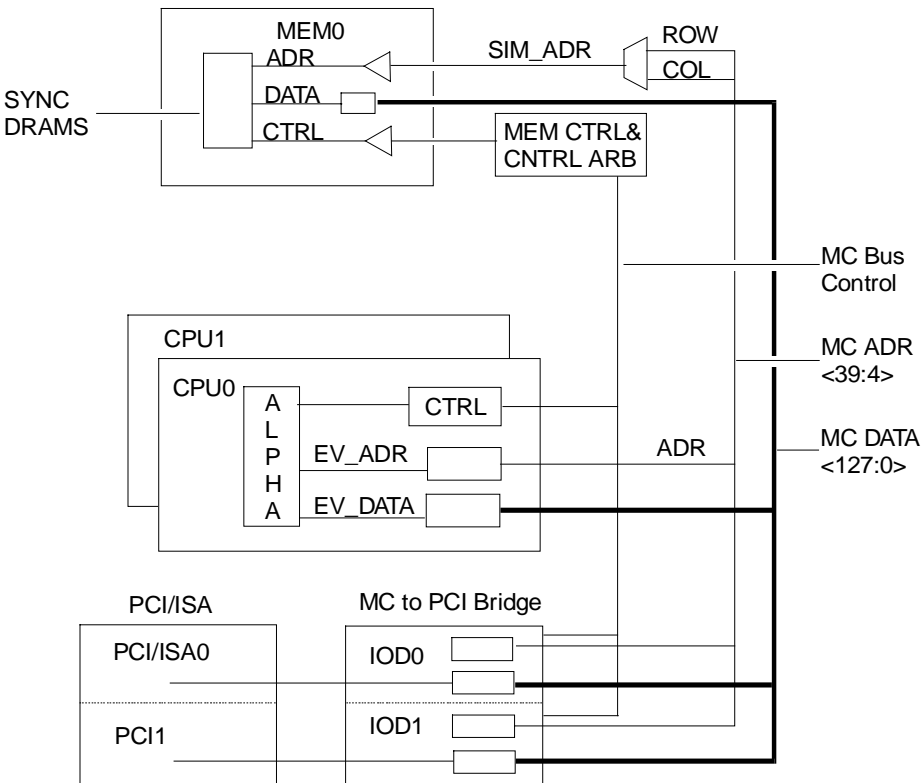
The five sections on the system motherboard are:

- The system bus or the CPU and memory backplane
- The power control logic
- The remote control logic
- The system bus to PCI bus bridges
- The PCI backplane containing two PCI buses, an EISA/ISA bus, a built-in CD-ROM controller, and an XBUS with several devices integral to the system.

1.8.1 System Bus (Backplane)

The system bus consists of a 40-bit command/address bus, a 128-bit plus ECC data bus, and several control signals and clocks. The system bus is part of the system motherboard.

Figure 1-9 System Bus Block Diagram



PKW0506-97

The system bus consists of a 40-bit command/address bus, a 128-bit plus ECC data bus, and several control signals, clocks, and a bus arbiter. The bus requires that all CPUs have the same high-speed oscillator providing the clock to the Alpha chip.

The system bus connects up to two CPUs, up to eight DIMM memory pairs on two riser cards, and two I/O bus bridges.

The system bus clock is provided by an oscillator on the CPU in slot CPU0. This oscillator is adjusted to maintain the system bus at a 66 MHz speed no matter what the speed of the CPU is.

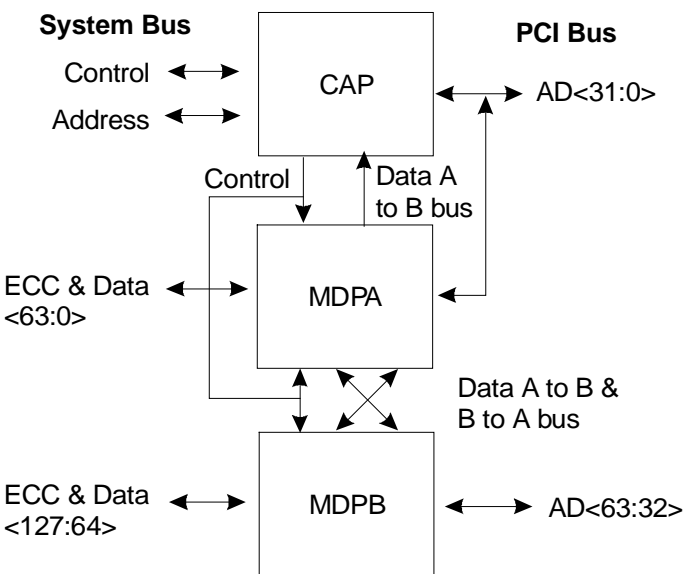
The system bus backplane initiates memory refresh transactions.

Five volt, 3.43 volt, and 12 volt power is provided directly to the motherboard from the power supplies.

1.8.2 System Bus to PCI Bus Bridge

The bridge is the physical interconnect between the system bus and the PCI bus.

Figure 1-10 System Bus to PCI Bus Bridge Block Diagram



PKW0507-97

The system bus to PCI bus bridge module converts system bus commands and data addressed to I/O space to PCI commands and data; and converts PCI bus commands and data addressed to system memory or CPUs to system bus commands and data.

The bridge has two major components:

- Command/address processor (CAP) chip
- Two data path chips (MDPA and MDPB)

There are two sets of these three chips, one set for each PCI.

The interface on the system bus side of the bridge responds to system bus commands addressed to the upper 64 Gbytes of I/O space. I/O space is addressed whenever bit <39> on the system bus address lines is set. The space so defined is 512 Gbytes in size. The first 448 Gbytes are reserved and the last 64 Gbytes, when bits <38:36> are set, are mapped to the PCI I/O buses.

The interface on the PCI side of the bridge responds to commands addressed to CPUs and memory on the system bus. On the PCI side, the bridge provides the interface to the PCIs. Each PCI bus is addressed separately. The bridge does not respond to devices communicating with each other on the same PCI bus. However, should a device on one PCI address a device on the other PCI bus, commands, addresses, and data run through the bridge out onto the system bus and back through the bridge to the other PCI bus.

In addition to its bridge function, the system bus to PCI bus bridge module monitors every transaction on the system bus for errors. It monitors the data lines for ECC errors and the command/address lines for parity errors.

1.8.3 PCI I/O Subsystem

The I/O subsystem consists of two 64-bit PCI buses. One has an embedded EISA/ISA bridge and three PCI option slots; the other has a built-in CD-ROM driver and three PCI option slots.

Figure 1-11 PCI Block Diagram

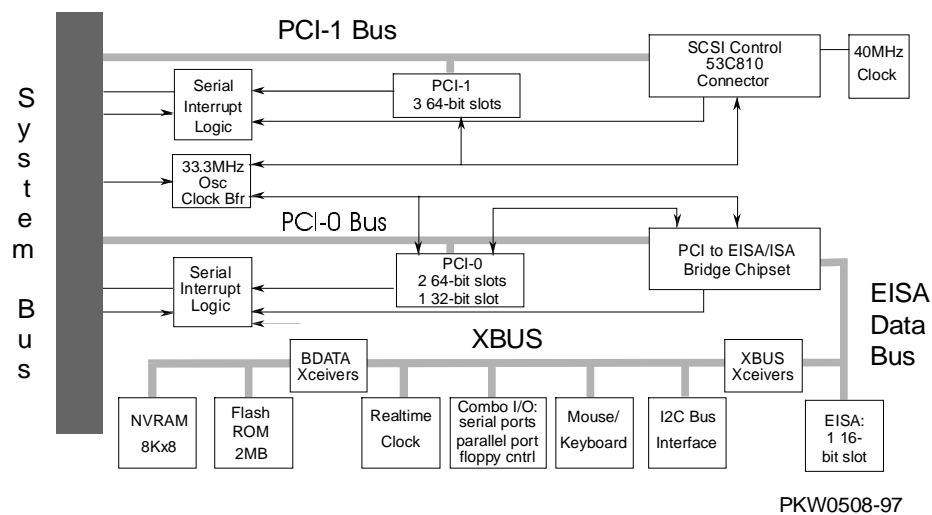


Table 1-1 PCI Motherboard Slot Numbering

Slot	PCI0	PCI1
1	PCI to EISA/ISA bridge	Internal CD-ROM controller
2	PCI slot	PCI slot
3	PCI slot	PCI slot
4	PCI slot	PCI slot

The logic for two PCI buses is on each PCI motherboard.

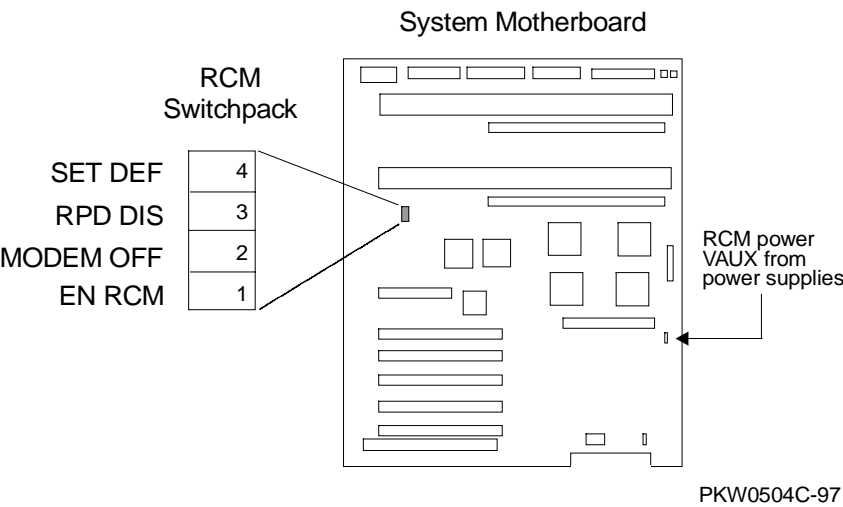
- PCI0 is a 64-bit bus with a built-in PCI to EISA/ISA bus bridge. PCI0 has three PCI slots and one EISA/ISA slot. When the EISA/ISA slot is used, PCI slot 4 on PCI bus 1 is not available. An 8-bit XBUS is connected to the EISA/ISA bus. On this bus there is an interface to the system F'C bus; mouse and keyboard support; an I/O combo controller supporting two serial ports, the floppy controller, and a parallel port; a real-time clock; two 1-Mbyte flash ROMs containing system firmware, and an 8-Kbyte NVRAM.
- PCI1 is a 64-bit bus with a built-in CD-ROM SCSI controller with three PCI slots.

Cable connectors to the CD-ROM, the floppy, and the OCP are on the motherboard. Connectors for the mouse, keyboard, two COM ports, the serial port, and a modem are on the system bulkhead. The bulkhead is part of the system motherboard.

1.8.4 Remote Control Logic

A section of the motherboard provides remote control operation of the system. A four-switch switchpack enables or disables remote control features.

Figure 1-12 Remote Control Logic



The system allows both local and remote control. A set of switches enables or disables remote control.

Table 1-2 Remote Control Switch Functions

Switch	Condition	Function
1 EN RCM	On (default)	Allows remote system control
	Off	Does not allow remote system control
2 Modem Off	On	Disables the RCM modem port
	Off (default)	Enable the RCM modem port
3 RPD DIS	On	Disables remote power down
	Off (default)	Enables remote power down
4 SET DEF	On	Resets the RCM microprocessor defaults
	Off (default)	Allows use of conditions set by the user

The default settings allow complete remote control. The user would have to change the switch settings to any other desired control.

See Appendix C for information on controlling the system remotely.

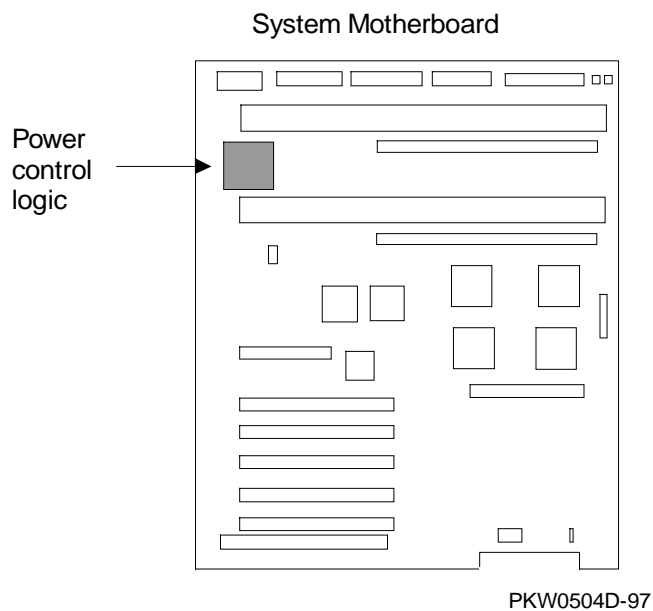
The remote console manager connects to a modem through the modem port on the bulkhead. The RCM uses VAUX power provided by the system power supplies.

The standard I/O ports (keyboard, mouse, COM1 and COM2 serial ports, and parallel ports) are on the same bulkhead.

1.8.5 Power Control Logic

The power control section of the motherboard controls power sequencing and monitors power supply voltage, system temperature, and fans.

Figure 1-13 Power Control Logic



The power control logic performs these functions:

- Monitors system temperature and powers down the system 30 seconds after it detects that internal temperature of the system is above the value of the environment variable **over_temp**. Default = 55^o C.
- Monitors the system and CPU fans at one second intervals and powers down the system 30 seconds after it detects a fan failure.
- Provides some visual indication of faults through LEDs.
- Controls reset sequencing.
- Provides I²C interface for fans, power supplies, and temperature signals:
 - Power supply 0, 1: present
 - Power supply 0, 1: power OK
 - CPU fan 0, 1: OK
 - CPU 1: present
 - Overtemp: Temp OK
 - System fan 0, 1: OK
 - Fan Kit OK

1.9 Power Circuit and Cover Interlock

Power is distributed throughout the system and mechanically can be broken by the On/Off switch, the cover interlock, or remotely through the RCM.

Figure 1-14 Power Circuit Diagram

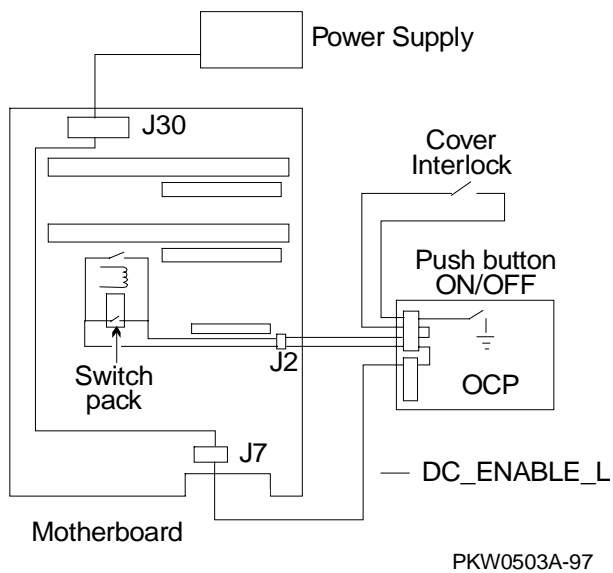


Figure 1-14 shows the distribution of power throughout the system. Opens in the circuit or the RCM signal RCM_DC_EN_L, or a power supply detected power fault interrupt DC power applied to the system. The opens can be caused by the On/Off button or the cover interlock.

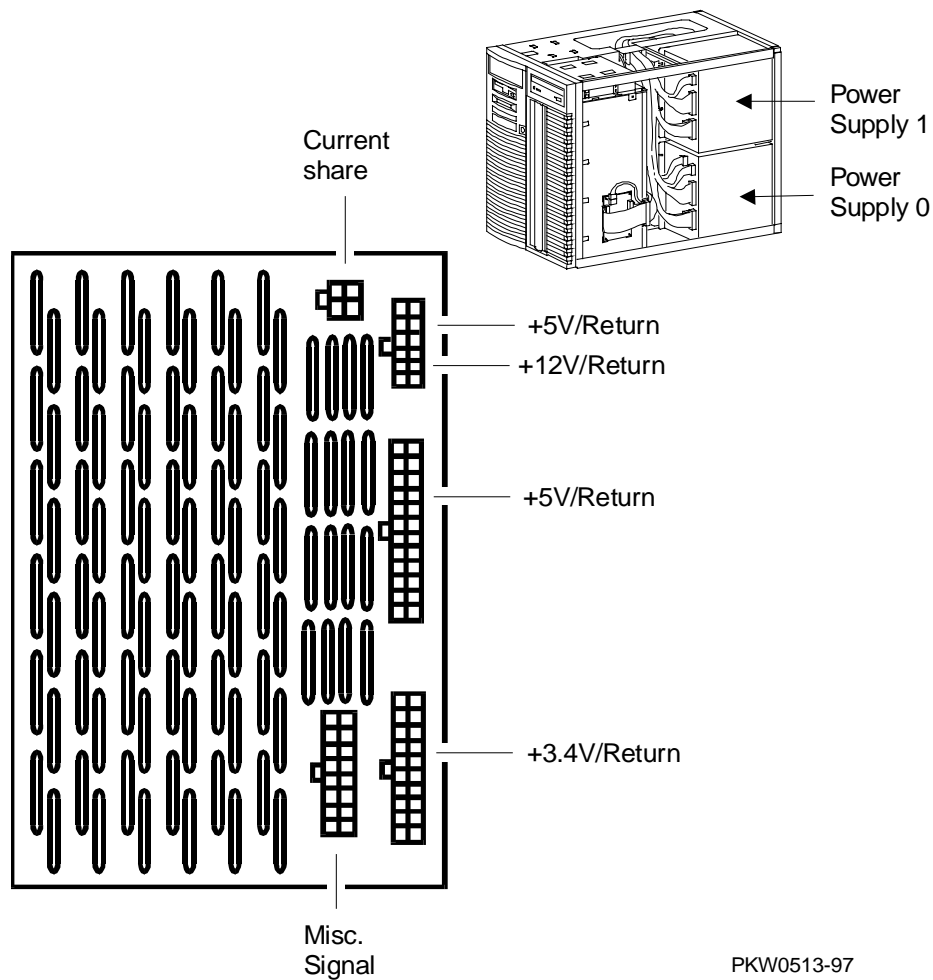
A failure anywhere in the circuit will result in the removal of DC power. A potential failure is the relay used in the remote control logic to control the RCM_DC_EN_L signal.

The cover interlock is located under the top cover between the system card cage and the storage area. To override the interlock, place a suitable object in the interlock switch that closes it.

1.10 Power Supply

Two power supplies provide system power.

Figure 1-15 Back of Power Supply and Location



Description

Two power supplies each provide 450 W to the system. Redundant power is not available at this time.

Power Supply Features

- 88–132 and 176–264 Vrms AC input
- 450 watts output. Output voltages are as follows:

Output Voltage	Min. Voltage	Max. Voltage	Max. Current
+5.0	4.90	5.25	52
+3.43	3.400	3.465	37.4
+12	11.5	12.6	17
–12	–13.2	–10.9	0.5
–5.0	–5.5	–4.6	0.2
Vaux	4.85	5.25	0.6

- Remote sense on +5.0V and +3.43V

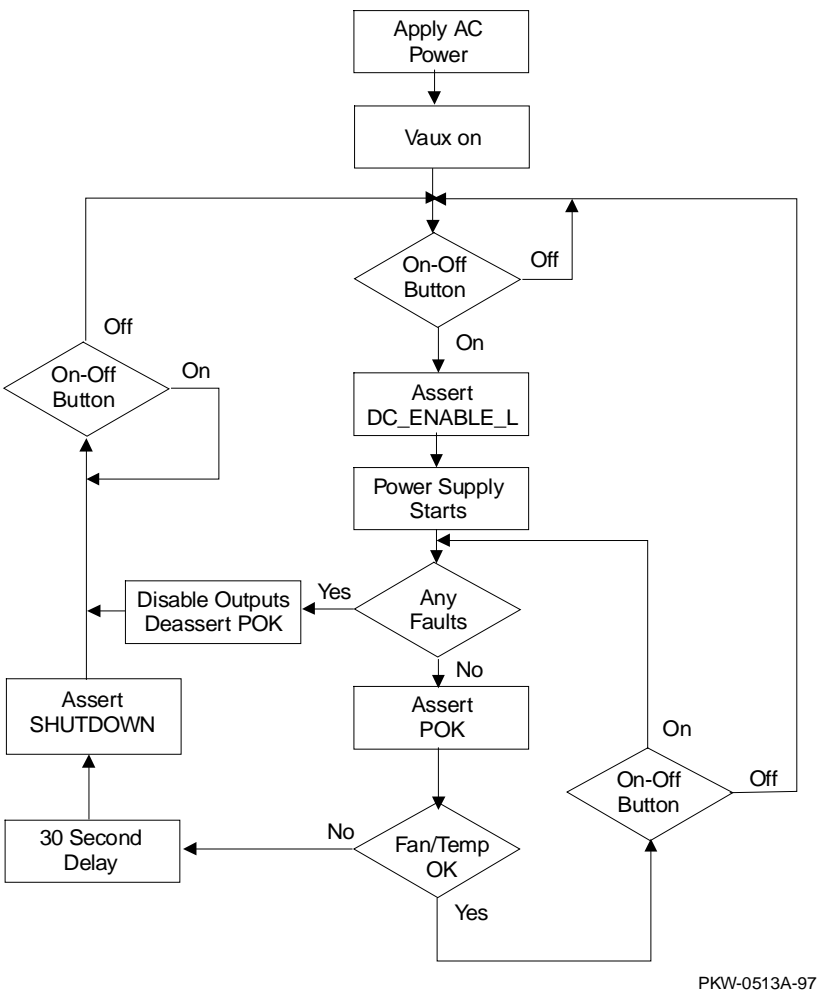
+5.0V is sensed on the system motherboard.
+3.43V is sensed on all CPUs in the system and the system bus motherboard.
- Current share on +5.0V, +3.43V, and +12V.
- 1 % regulation on +3.43V.
- Fault protection (latched). If a fault is detected by the power supply, it will shut down. The power supply faults detected are:

Fan Failure
Over-voltage
Overcurrent
Power overload
- DC_ENABLE_L input signal starts the DC outputs.
- SHUTDOWN_H input signal shuts the power supply off in case of a system fan or CPU fan failure.
- POK_H output signal indicates that the power supply is operating properly.

1.11 Power Up/Down Sequence

System power can be controlled manually by the On/Off button on the OCP or remotely through the RCM. The power-up/down sequence flow is shown below.

Figure 1-16 Power Up/Down Sequence Flowchart



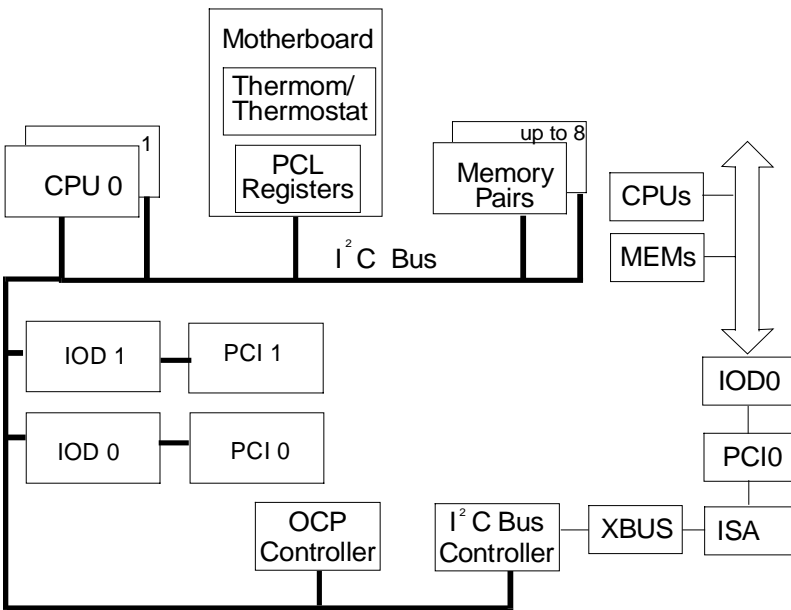
PKW-0513A-97

When AC is applied to the system, Vaux (auxiliary voltage) is asserted and is sensed by the power control logic (PCL) section of the motherboard if the On-Off Button is On. The PCL asserts DC_ENABLE_L starting the power supplies. If there is a hard fault on power-up, the power supplies shut down immediately; otherwise, the power system powers up and remains up until the system is shut off or the PCL senses a fault. If a power fault is sensed, the signal SHUTDOWN is asserted after a 30 second delay. Cycling the On-Off button can restore the power.

1.12 Maintenance Bus (I²C Bus)

The I²C bus (referred to as the “I squared C bus”) is a small internal maintenance bus used to monitor system conditions scanned by the power control logic, write the fault display, store error state, and track configuration information in the system. Although all system modules (not I/O modules) sit on the maintenance bus, only the I²C controller accesses it.

Figure 1-17 I²C Bus Block Diagram



PKW0511-97

Monitor

The I²C bus monitors the state of system conditions scanned by the power control logic. There are two registers that the PC logic writes data to:

- One records the state of the fans and power supplies and is latched when there is a fault.
- The other causes an interrupt on the I²C bus when a CPU or system fan fails, an overtemperature condition exists, or power supplied to the system exhibits an overcurrent condition.

The interrupt received by the I²C bus controller on PCI 0 and passed on to the IOD 0 chip set alerts the system of imminent power shutdown. The controller has 30 seconds to read the two registers and store the information in the EEPROM on the motherboard. The SRM console command **show power** reads these registers.

Fault Display

The OCP display is written through the I²C bus.

Error State

Error state is stored for power, fan, and overtemperature conditions on the I²C bus.

Configuration Tracking

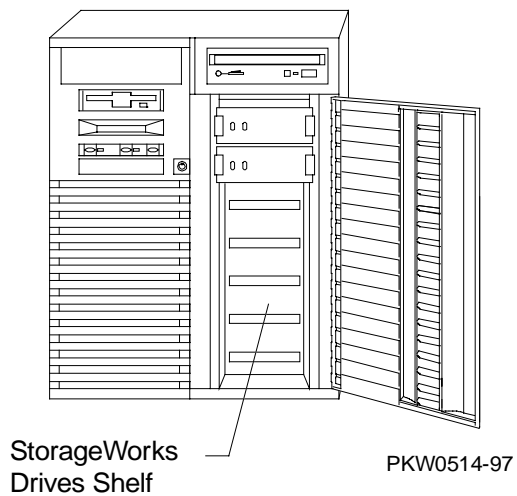
Each CPU and each logical section of the system motherboard (the PCI bridge, the PCI backplane, the power control logic, the remote console manager), and the system motherboard itself has an EEPROM that contains information about the module that can be written and read over the I²C bus. All EEPROMs contain the following information:

- Module type
- Module serial number
- Hardware revision for the logical block
- Firmware revision

1.13 StorageWorks Drives

The system supports up to seven StorageWorks drives.

Figure 1-18 StorageWorks Drive Location



The StorageWorks drives are to the right of the system cage. Up to seven drives fit into the shelf. The system supports fast wide Ultra SCSI disk drives. The RAID controller is also supported. With an optional Ultra SCSI Bus Splitter Kit the StorageWorks shelf can be split into two buses.

Chapter 2

Power-Up

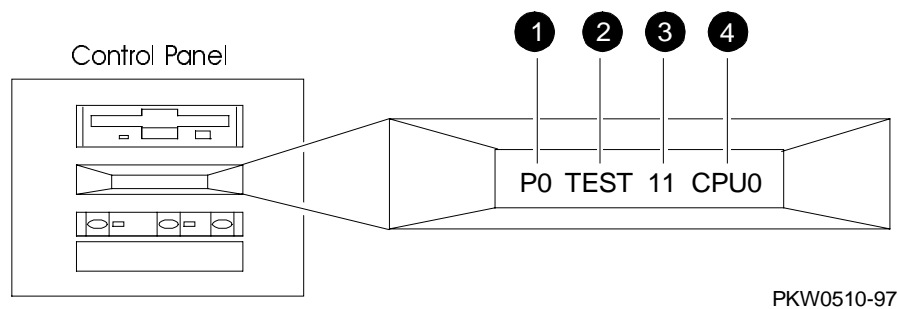
This chapter describes system power-up testing and explains the power-up displays. The following topics are covered:

- Control Panel
- Power-Up Sequence
- SRAM Power-Up Test Flow
- SRAM Errors Reported
- XSRAM Power-Up Test Flow
- XSRAM Errors Reported
- Console Power-Up Tests
- Console Device Determination
- Console Power-Up Display
- Fail-Safe Loader

2.1 Control Panel

The control panel display indicates the likely device when testing fails.

Figure 2-1 Control Panel and LCD Display



PKW0510-97

- When the On/Off button LED is on, power is applied and the system is running. When it is off, the system is not running, but power may or may not be present. If the power supplies are receiving AC power, Vaux is present on the system motherboard regardless of the condition of the On/Off switch.
- When the Halt button LED is lit and the On/Off button LED is on, the system should be running either the SRM console or Windows NT.

Table 2-1 Control Panel Display

Field	Content	Display	Meaning
❶	CPU number	P0–P1	CPU reporting status
❷	Status	TEST	Tests are executing
		FAIL	Failure has been detected
		MCHK	Machine check has occurred
		INTR	Error interrupt has occurred
❸	Test number		
❹	Suspected device	CPU0–1	CPU module number
		MEM0–7 and L, H, or *	Memory pair number and low DIMM, high DIMM, or either
		IOD0	Bridge to PCI bus 0 ¹
		IOD1	Bridge to PCI bus 1 ¹
		FROM0	Flash ROM ¹
		COMBO	COM controller ¹
		PCEB	PCI-to-EISA bridge ¹
		ESC	EISA system controller ¹
		NVRAM	Nonvolatile RAM ¹
		TOY	Real-time clock ¹
		I8242	Keyboard and mouse controller ¹

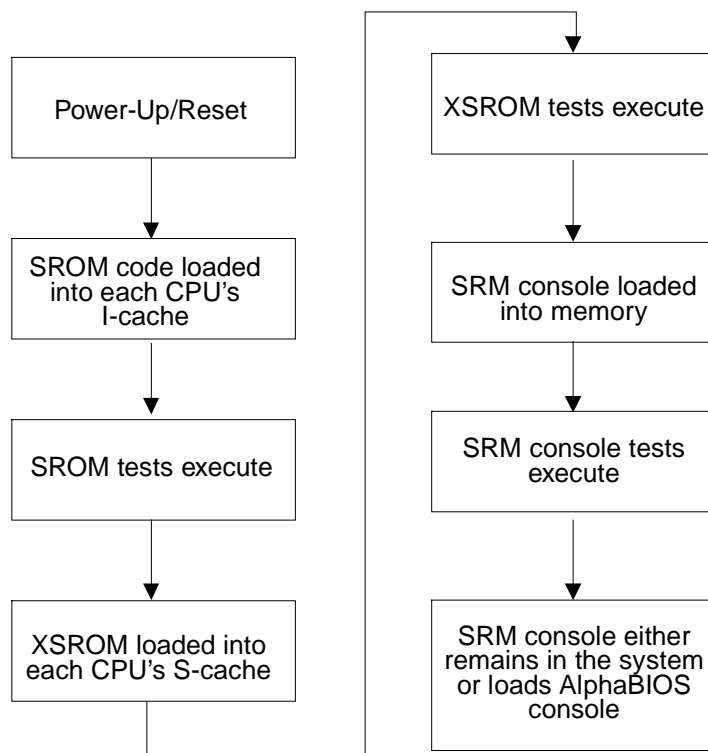
The potentiometer, accessible through the access hole just above the Reset button controls the intensity of the LCD. Use a small Phillips head screwdriver to adjust.

¹ On the system motherboard (54-25147-01)

2.2 Power-Up Sequence

Console and most power-up tests reside on the I/O subsystem, not on the CPU nor on any other module on the system bus.

Figure 2-2 Power-Up Flow



PKW0432B-96

Definitions

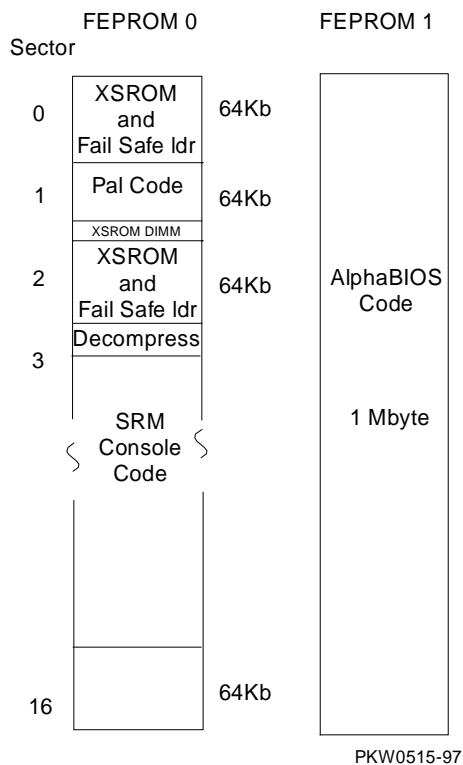
SRM. The SRM is a 128-Kbit ROM on each CPU module. The ROM contains minimal diagnostics that test the Alpha chip and the path to the XSROM. Once the path is verified, it loads XSROM code into the Alpha chip and jumps to it.

XSROM. The XSROM, or extended SRM, contains back-up cache and memory tests, the I/O subsystem tests for embedded devices, and a fail-safe loader. The

XSROM code resides in sector 0 of FEPROM 0 on the XBUS. Sector 2 of FEPROM 0 contains a duplicate copy of the code and is used if sector 0 is corrupt. Code for sizing DIMM memory resides in sector 1 of FEPROM 0 along with the PAL code.

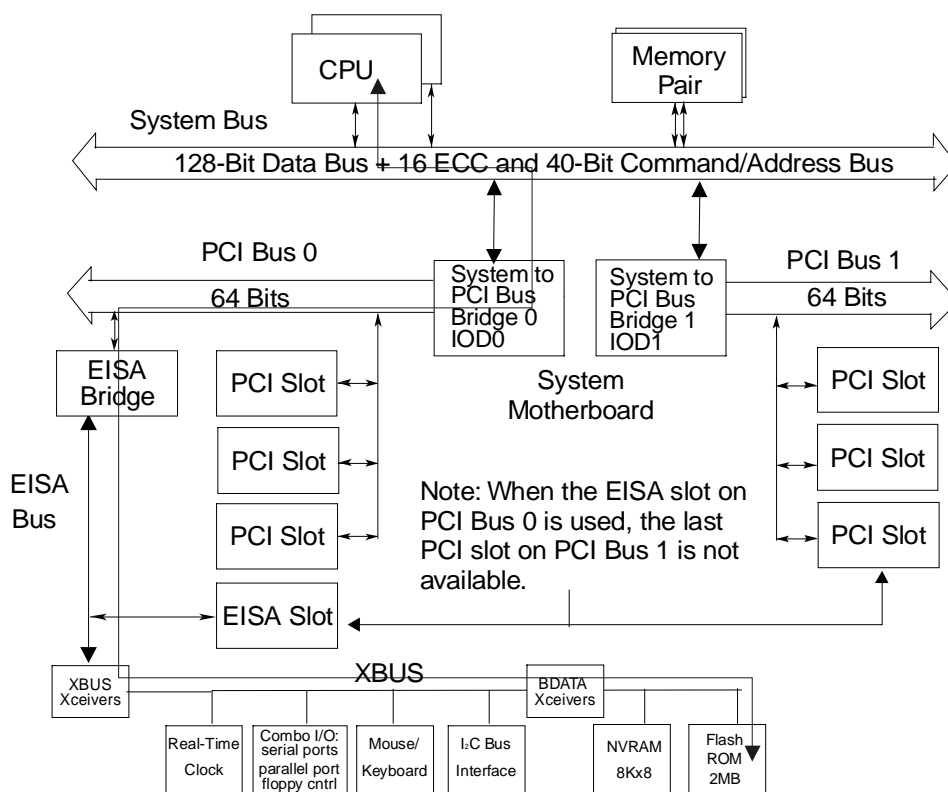
FEPROM. Two 1-Mbyte programmable ROMs (FEPROMS) are on the XBUS on PCI0. FEPROM 0 contains two copies of the XSROM, the OpenVMS and DIGITAL UNIX PAL code, and the SRM console and decompression code. FEPROM 1 contains the AlphaBIOS and NT HAL code. See Figure 2-3. These two FEPROMs can be flash updated. Refer to Appendix A.

Figure 2-3 Contents of FEPROMs



For the console to run, the path from the CPU to the XSROM must be functional. The XSROM resides in FEPR0M0 on the XBUS, off the EISA bus, off PCI 0, off IOD 0. See Figure 2-4. This path is minimally tested by SROM.

Figure 2-4 Console Code Critical Path (Block Diagram)



PKW0502A-97

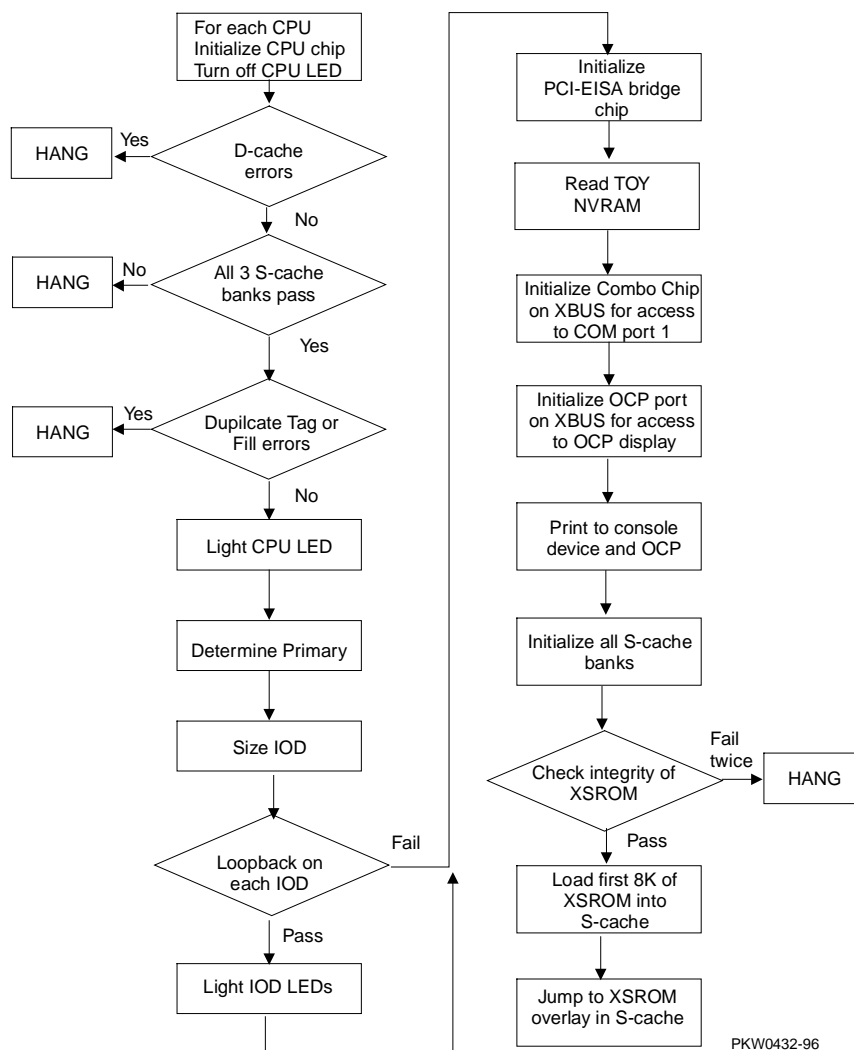
The SRM contents are loaded into each CPU's I-cache and executed on power-up/reset. After testing the caches on each processor chip, it tests the path to the XSROM. Once this path is tested and deemed reliable, layers of the XSROM are loaded sequentially into the processor chip on each CPU. None of the SRM or XSROM power-up tests are run from memory—all run from the caches in the CPU chip, thus providing excellent diagnostic isolation. Later power-up tests, run under the console, are used to complete testing of the I/O subsystem.

There are two console programs: the SRM console and the AlphaBIOS console, as detailed in your system *User's Guide*. By default, the SRM console is always loaded and I/O system tests are run under it before the system loads AlphaBIOS. To load AlphaBIOS, the **os_type** environment variable must be set to **NT** and "halt assertion" must be clear. Otherwise, the SRM console continues to run.

2.3 SROM Power-Up Test Flow

The SROM tests the CPU chip and the path to the XSROM.

Figure 2-5 SROM Power-Up Test Flow



PKW0432-96

The Alpha chip built-in self-test tests the I-cache at power-up and upon reset.

Each CPU chip loads its SRAM code into its I-cache and starts executing it. If the chip is partially functional, the SRAM code continues to execute. However, if the chip cannot perform most of its functions, that CPU hangs and that CPU pass/fail LED remains off. (In these systems, the CPU pass/fail LED is not visible.)

If the system has more than one CPU and at least one passes both the SRAM and XSROM power-up tests, the system will bring up the console. The console checks the FW_SCRATCH register where evidence of the power-up failure is left. Upon finding the error, the console sends these messages to COM1 and the OCP:

- COM1 (or VGA): Power-up tests have detected a problem with your system
- OCP: Power-up failure

Table 2-2 lists the tests performed by the SROM.

Table 2-2 SROM Tests

Test Name	Logic Tested
D-cache RAM March test	D-cache access, D-cache data, D-cache address logic
D-cache Tag RAM March test	D-cache tag store RAM, D-cache bank address logic
S-cache Data March test	S-cache RAM cells, S-cache data path, S-cache address path
S-cache Tag RAM March test	S-cache tag store RAM, S-cache bank address logic
I-cache Parity Error test	I-cache parity error detection, ISCR register and error forcing logic, IC_PERR_STAT register and reporting logic
D-cache Parity Error test	D-cache parity error detection, DC_MODE register and parity error forcing logic, DC_PERR_STAT register and reporting logic
S-cache Parity Error test	S-cache parity error detection, SC_CTL register and parity error forcing logic, SC_STAT register and reporting logic
IOD Access test	Access to IOD CSRs, data path through CAP chip and MDP0 on each IOD, PCIO A/D lines <31:0>

2.4 SROM Errors Reported

The SROM reports machine checks, pending interrupt/exception errors, and errors related to corruption of FEPROM 0. If SROM errors are fatal, the particular CPU will hang and only the CPU self-test pass LEDs and/or the LEDs on the system motherboard will indicate the failure. The CPU self-test pass LED is not visible but the IOD0 and IOD1 pass LEDs are.

Example 2-1 SROM Errors Reported at Power-Up

Unexpected Machine Check (CPU Error)

```
UNEX MCHK on CPU 0
EXC_ADR 42a9
EI_STAT ffffffff004fffffff
EI_ADDR fffffff000000801f
SC_STAT 0
SC_ADDR FFFFFFF0000005F2F
```

Pending Interrupt/Exception (CPU Error)

```
INT-EXC on CPU0
ISR          400000
EI_STAT ffffffff007fffffff
EI_ADDR fffffff7ffffffffffdf
FIL_SYN 631B
BCTGADR fffffffa7ffffcafff
```

FEPROM Failures (PCI Motherboard Error)

Sector 0 failures (XSROM flash unload failure)

```
Sctr 0 -XSROM headr PTTRN fail
Sctr 0 -XSROM headr CHKSM fail
Sctr 0 -XSROM code  CHKSM fail
```

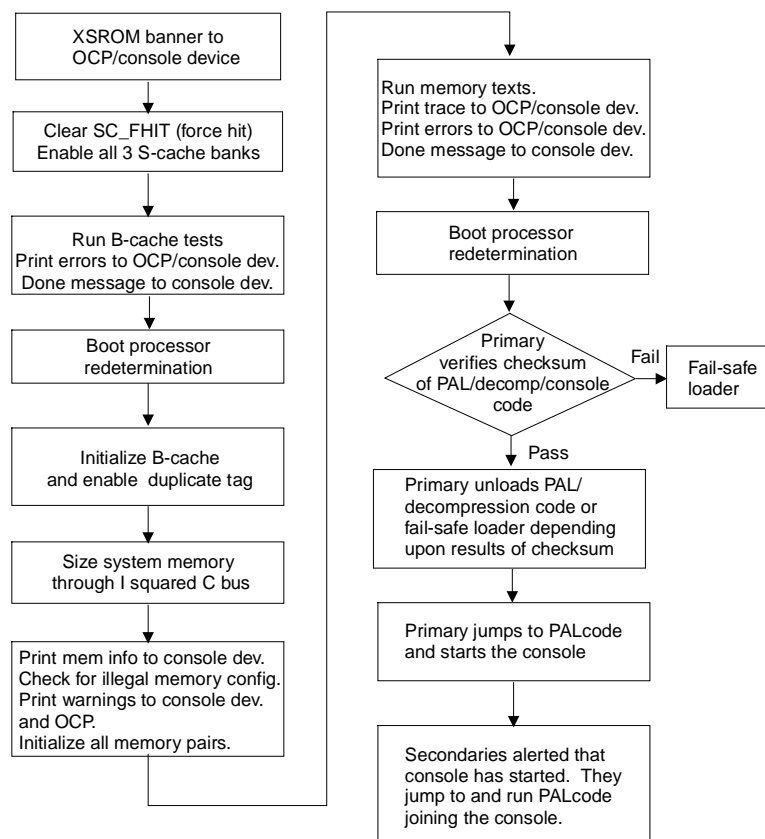
Sector 2 failures (XSROM recovery flash unload failure)

```
Sctr 2 -XSROM headr PTTRN fail
Sctr 2 -XSROM headr CHKSM fail
Sctr 2 -XSROM code  CHKSM fail
```

2.5 XSROM Power-Up Test Flow

Once the SROM has completed its tests and verified the path to the FEPRM containing the XSROM code, it loads the first 8 Kbytes of XSROM into the primary CPU's S-cache and jumps to it.

Figure 2-6 XSROM Power-Up Flowchart



Note: The XSROM can only print to the console device if the environment variable console = serial. It always sends output to the OCP.

PKW0432A-96

XSROM tests are described in Table 2-3. Failure indicates a CPU failure.

After jumping to the primary CPU's S-cache, the code then intentionally I-caches itself and is completely register based (no D-stream for stack or data storage is used). The only D-stream accesses are writes/reads during testing.

Each FEPROM has sixteen 64-Kbyte sectors. The first sector contains B-cache tests, memory tests, and a fail-safe loader. The second sector contains support for system memory and PALcode. The third sector contains a copy of the first sector. The remaining thirteen sectors contain the SRM console and decompression code.

NOTE: Memory tests are run during power-up and reset (see Table 2-4). They are also affected by the state of the memory_test environment variable, which can have the following values:

<i>FULL</i>	Test all memory
<i>PARTIAL</i>	Test up to the first 256 Mbytes
<i>NONE</i>	Test 32 Mbytes

Table 2-3 XSROM Tests

Test	Test Name	Logic Tested
11	B-cache Data March test	B-cache data RAMs, CPU chip B-cache control, CPU chip B-cache address decode, INDEX_H<23:6> (address bus)
12	B-cache Tag March test	B-cache tag store RAMs, B-cache STAT store RAMs
13	B-cache ECC Data Line test	CPU chip ECC generation and checking logic, ECC lines from CPU chip to B-cache, B-cache ECC RAMs
14	B-cache Tag Data Line test	Access to B-cache tags, shorts between tag data and its status and parity bits
15	B-cache Data Line test	B-cache data lines to B-cache data RAMs, B-cache read/write logic
16	B-cache ECC Data Line test	CPU chip ECC generation and checking logic, ECC lines from CPU chip to B-cache, B-cache ECC RAMs

Table 2-4 Memory Tests

Test	Test Name	Logic Tested	Description
20	Memory Data test	Data path to and from memory Data path on memory and RAMs	Test floats 1 and 0 across data and check bit data lines. Errors are reported for each DIMM memory card from MEM0_L to MEM7_H.
21	Memory Address test	Address path to and from memory Address path on memory and RAMs	Same as test 20.
23*	Memory Bitmap Building	No new logic	Maps out bad memory by way of the bitmap. It does not completely fail memory.
24	Memory March test	No new logic	Maps out bad memory.

* There is no test 22.

2.6 XSROM Errors Reported

The XSROM reports B-cache test errors and memory test errors. It also reports a warning if memory is illegally configured.

Example 2-2 XSROM Errors Reported at Power-Up

B-cache Error (CPU Error)

```
TEST ERR on cpu0                #CPU running the test
FRU      cpu0
err#    2
tst#    11
exp:    5555555555555555        #Expected data
rcv:    aaaaaaaaaaaaaaaaaa      #Received data
adr:    ffff8                   #B-cache location
                                #error occurred
```

Memory Error (Memory Module Indicated)

```
20..21..
TEST ERR on cpu0                #CPU running test
FRU:    MEM1L                   #Low member of memory pair 1

err#    c
tst#    21
22..23..24..Memory testing complete on cpu0
```

Memory Configuration Error (Operator Error)

```
ERR!    mem_pair0 misconfigured
ERR!    mem_pair1 card size mismatch
ERR!    mem_pair6 card type mismatch
ERR!    mem_pair1 EMPTY
```

FEPROM Failures (PCI Error)

```
Sctr 1 -PAL headr PTTRN fail
Sctr 1 -PAL headr CHKSM fail
Sctr 1 -PAL code  CHKSM fail
Sctr 3 -CONSLE headr PTTRN fail
Sctr 3 -CONSLE headr CHKSM fail
Sctr 3 -CONSLE code  CHKSM fail
```

2.7 Console Power-Up Tests

Once the SRM console is loaded, it tests each IOD further. Table 2-5 describes the IOD power-up tests, and Table 2-6 describes the PCI power-up tests.

Table 2-5 IOD Tests

Test #	Test Name	Description
1	IOD CSR Access test	Read and write all CSRs in each IOD.
2	Loopback test	Dense space writes to the IOD's PCI dense space to check the integrity of ECC lines.
3	ECC test	Loopback tests similar to test 2 but with a varying pattern to create an ECC of 0s. Single- and double-bit errors are checked.
4	Parity Error and Fill Error tests	Parity errors are forced on the address and data lines on system bus and PCI buses. A fill error transaction is forced on the system bus.
5	Translation Error test	A loopback test using scatter/gather address translation logic on each IOD.
6	Write Pending test	Runs test 2 with the write-pending bit set and clear in the CAP chip control register.
7	PCI Loopback test	Loops data through each PCI on each IOD, testing the mask field of the system bus.
8	PCI Peer-to-Peer Byte Mask test	Tests that devices on the same PCI and on different PCIs can communicate.
9 ¹	Page Table Entry test 1 (CAP chip)	Tests every PTE using scatter/gather translation and addressing.
10 ¹	Page Table Entry test 2 (CAP chip)	Tests random PTEs forcing use of all interesting tag and page registers.

¹ Not run on power-up. These tests take approximately 30 seconds and are run in user mode.

Table 2-6 PCI Motherboard Tests

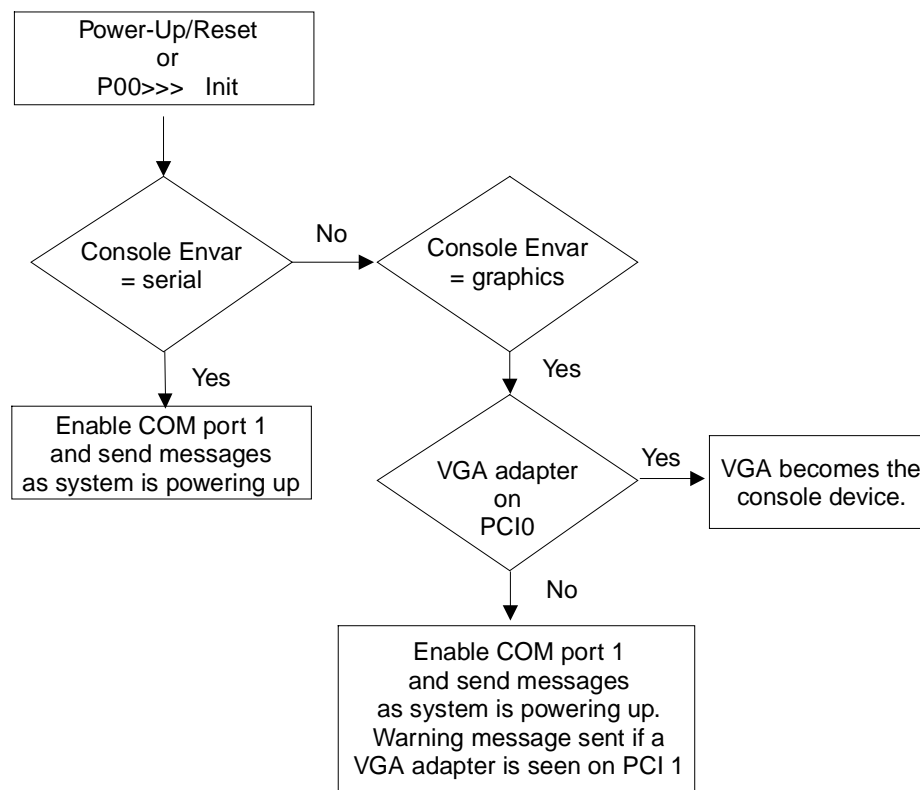
Test Number	Test Name	Diagnostic Name	Description
1	PCEB	pceb_diag	Tests the PCI to EISA bridge chip
2	ESC	esc_diag	Tests the EISA system controller
3	8K NVRAM	nvram_diag	Tests the NVRAM
4	Real-Time Clock	ds1287_diag	Tests the real-time clock chip
5	Keyboard and Mouse	i8242_diag	Tests the keyboard/mouse chip
6	Flash ROM	flash_diag	Dumps contents of flash ROM
7	Serial and Parallel Ports and Floppy	combo_diag	Tests COM ports 1 and 2, the parallel port, and the floppy
8	CD-ROM	ncr810_diag	Tests the CD-ROM controller

For both IOD tests and PCI 0 and PCI 1 tests, trace and failure status is sent to the OCP. If any of these tests fail, a warning is sent to the SRM console device after the console prompt (or AlphaBIOS pop-up box). The IOD LEDs on the system motherboard are controlled by the diagnostics. If a LED is off, a failure occurred.

2.8 Console Device Determination

After the SROM and XSROM have completed their tasks, the SRM console program, as it starts, determines where to send its power-up messages.

Figure 2-7 Console Device Determination Flowchart



PKW0434-96

Console Device Options

The console device can be either a serial terminal or a graphics monitor. Specifically:

- A serial terminal connected to COM1 off the bulkhead. The terminal connected to COM1 must be set to 9600 baud. This baud rate cannot be changed.
- A graphics monitor off an adapter on PCIO.

Systems running Windows NT must have a graphics monitor as the console device and run AlphaBIOS as the console program.

During power-up, the SRM and the XSROM always send progress and error messages to the OCP and to the COM1 serial port if the SRM **console** environment variable (set with the **set console** command) is set to **serial**. If the **console** environment variable is set to **graphics**, no messages are sent to COM1.

If the console device is connected to COM1, the SRM, XSROM, and console power-up messages are sent to it once it has been initialized. If the console device is a graphics device, console power-up messages are sent to it, but SRM and XSROM power-up messages are lost. No matter what the **console** environment variable setting, each of the three programs sends messages to the control panel display.

Messages Sent By	Console Set to Serial	Graphics
SRM	COM1	Lost, though a subset is sent to the OCP
XSROM	COM1	Lost, though a subset is sent to the OCP
SRM console	COM1	VGA, though a subset is sent to the OCP

Changing Where the Console Output Is Displayed

You can change where console output is displayed, assuming the SRM console has fully powered up and the **os_type** environment variable is set to **openvms** or **unix**. (The following does not work if **os_type** is set to **nt**.)

If the **console** environment variable is set to **serial** and no serial terminal is attached to COM1, pressing a carriage return on a graphics monitor attached to the system makes it the console device and the console prompt is sent to it. If the console environment variable is set to **graphics** and no graphics monitor is attached to the adapter, pressing a carriage return on a serial terminal attached to COM1 makes it the console device and the console prompt is sent to it. In either case power-up information is lost.

2.9 Console Power-Up Display

The entire power-up display prints to a serial terminal (if the *console* environment variable is set to *serial*), and parts of it print to the control panel display. The last several lines print to either a serial terminal or a graphics monitor.

Example 2-3 Power-Up Display

SROM V3.0 on cpu0	❶
SROM V3.0 on cpu1	
XSROM V5.0 on cpu0	❷
XSROMb V5.0 on cpu1	
BCache testing complete on cpu1	❸
BCache testing complete on cpu0	
mem_pair0 - 256 MB	❹
mem_pair1 - 256 MB	
mem_pair2 - 64 MB	
mem_pair3 - 64 MB	
20..21..20..21..23..24..24..	❺
Memory testing complete on cpu0	
Memory testing complete on cpu1	

- ❶ At power-up or reset, the SROM code on each CPU module is loaded into that module's I-cache and tests the module. If all tests pass, the processor's LED lights. If any test fails, the LED remains off and power-up testing terminates on that CPU.

The first determination of the primary processor is made, and the primary processor executes a loopback test to each PCI bridge. If this test passes, the bridge LED lights. If it fails, the LED remains off and power-up continues. The EISA system controller, PCI-to-EISA bridge, COM1 port, and control panel port are all initialized thereafter.

Each CPU prints an SROM banner to the device attached to the COM1 port and to the control panel display. (The banner prints to COM1 if the **console** environment variable is set to **serial**. If it is set to **graphics**, nothing prints to the console terminal, only to the control panel display, until ❹ occurs).

- ❷ Each processor's S-cache is initialized, and the XSROM code in the FEPROM on the PCI 0 is unloaded into them. (If the unload is not successful, a copy is unloaded from a different FEPROM sector. If the second try fails, the CPU hangs.)

Each processor jumps to the XSROM code and sends an XSROM banner to the COM1 port and to the control panel display.

- ❸ The three S-cache banks on each processor are enabled, and then the B-cache is tested. If a failure occurs, a message is sent to the COM1 port and to the control panel display.

Each CPU sends a B-cache completion message to COM1.

- ❹ The primary CPU is again determined, and memory is sized using code in sector 1 of FEPROM 0.

The information on memory pairs is sent to COM1. If an illegal memory configuration is detected, a warning message is sent to COM1 and the control panel display.

- ❺ Memory is initialized and tested, and the test trace is sent to COM1 and the control panel display. Each CPU participates in the memory testing. The numbers for tests 20 and 21 might appear interspersed, as in Example 2–3. This is normal behavior. Test 24 can take several minutes if the memory is very large. The message "P0 TEST 24 MEM**" is displayed on the control panel display; the second asterisk rotates to indicate that testing is continuing. If a failure occurs, a message is sent to the COM1 port and to the control panel display.

Each CPU sends a test completion message to COM1.

Continued on next page

Example 2-3 Power-Up Display (Continued)

```
starting console on CPU 0 ⑥
sizing memory ⑦
  0      256 MB DIMM
  1      256 MB DIMM
64 MB DIMM
64 MB DIMM
starting console on CPU 1
probing IOD1 hose 1 ⑧
  bus 0 slot 1 - NCR 53C810
  bus 0 slot 2 - DECchip 21041-AA
  bus 0 slot 3 - NCR 53C810
probing IOD0 hose 0
  bus 0 slot 1 - PCEB
    probing EISA Bridge, bus 1
  bus 0 slot 2 - S3 Trio64/Trio32
  bus 0 slot 3 - DECchip 21140-AA
Configuring I/O adapters...
  Ncr0, hose 1, bus 0, slot 1
  Tulip0, hose 1, bus 0, slot 2
  Ncr1, hose 1, bus 0, slot 3
  Floppy0, hose 0, bus 1 slot 0
  Mc0, hose 0 bus 0, slot 2
  tulip1, hose 0, bus 0, slot 3
System temperature is 31 degrees C
AlphaServer 1200 Console V5.0, 02-SEP-1997 18:18:26 ⑨
P00>>>
```

- ⑥ The final primary CPU determination is made. The primary CPU unloads PALcode and decompression code from the FEPROM on PCI 0 to its B-cache. The primary CPU then jumps to the PALcode to start the SRM console.

The primary CPU prints a message indicating that it is running the console. Starting with this message, the power-up display is printed to the default console terminal, regardless of the state of the **console** environment variable. (If **console** is set to **graphics**, the display from here to the end is saved in a memory buffer and printed to the graphics monitor after the PCI buses are sized and the graphics device is initialized.)

- ⑦ The size and type of each memory pair is determined.

The console is started on each of the secondary CPUs. A status message prints for each CPU.

- ⑧ The PCI bridges (indicated as IOD n) are probed and the devices are reported. I/O adapters are configured.

- ⑨ The SRM console banner and prompt are printed. (The SRM prompt is shown in this manual as P00>>>. It can, however, be P01>>>.) If the **auto_action** environment variable is set to **boot** or **restart** and the **os_type** environment variable is set to **unix** or **openvms**, the DIGITAL UNIX or OpenVMS operating system boots.

If the system is running the Windows NT operating system (the **os_type** environment variable is set to **nt**), the SRM console loads and starts the AlphaBIOS console and does not print the SRM banner or prompt.

2.10 Fail-Safe Loader

The fail-safe loader is a software routine that loads the SRM console image from floppy. Once the console is running you will want to run LFU to update FEPROM 0 with a new image.

NOTE: FEPROM 0 contains images of the SROM, XSROM, PAL, decompression, and SRM console code.

If the fail-safe loader loads, the following conditions exist on the machine:

- The SROM has passed its tests and successfully unloaded the XSROM. If the SROM fails to unload both copies of XSROM, it reports the failure to the control panel display and COM1 if possible, and the system hangs.
- The XSROM has completed its B-cache and memory tests but has failed to unload the PALcode in FEPROM 0 sector 1 or the SRM console code.
- The XSROM reports the errors encountered and loads the fail-safe loader.

Chapter 3

Troubleshooting

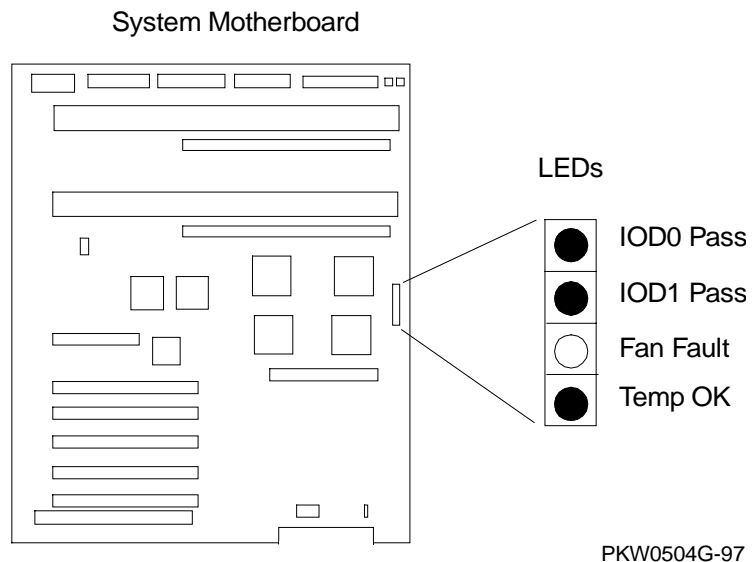
This chapter describes troubleshooting during power-up and booting. It also describes the console test command and other useful commands. The following topics are covered:

- Troubleshooting with LEDs
- Troubleshooting Power Problems
- Running Diagnostics—Test Command
- Releasing Secure Mode
- Testing an Entire System
- Other Useful Console Commands

3.1 Troubleshooting with LEDs

During power-up, reset, initialization, or testing, diagnostics are run on CPUs, memories, I/O bridges, and the PCI backplane and its embedded options. This section describes possible problems that can be identified by checking LEDs. Unfortunately LEDs on the CPU module are not visible; the only visible LEDs are on the system motherboard.

Figure 3-1 System Motherboard LEDs



System Motherboard LEDs

You see the system motherboard LEDs by looking through the grate at the back of the machine. The normal state of the LEDs is shown in Figure 3-1.

- If one of the IOD LEDs is off, the system bus to PCI bus bridge has failed. Replace the system motherboard.
- If the Fan Fault LED is ON, at least one of the four fans is broken. If this condition occurs while the system is up and running, an error message identifying the FRU is printed to the console. If this condition occurs during a cold start, identifying which fan caused the fan fault depends upon which type of console the system has. If your console is a serial terminal (for OpenVMS or DIGITAL UNIX), the error identifying which fan failed is reported at the console. If your console is a graphics monitor (for NT), reset the system and watch the OCP display. During the first 30 seconds, one of the following message should occur:
 - SYSx Fan Failed where x = 0 or 1
 - CPUx Fan Failed where x = 0 or 1

Replace the failing FRU.

- If the Temp OK LED is OFF, an overtemperature condition exists. Several things can cause this condition: blocked airflow, temperature in the room where the system is located is too high, the system card cage is open and air is not channeled properly over the system. Fix any of these conditions, if possible. The overtemperature threshold is programmable and is controlled by the environment variable **over_temp**. Its default is 55 degrees C. After the system has cooled down and can be powered up, you can change the threshold. If you do this and the temperature inside the system gets too hot, it is likely that system errors will occur and the system may crash.

3.2 Troubleshooting Power Problems

Power problems can occur before the system is up or while the system is running.

Power Problem List

The system will halt for the following reasons:

1. A CPU fan failure
2. A system fan failure
3. An overtemperature condition
4. Power supply failure
5. Circuit breaker(s) tripped
6. AC problem
7. Interlock switch activation or failure
8. Environmental electrical failure or unrecoverable system fault with auto_action ev = halt or boot
9. Cable failure

Indication of failure:

1. LEDs indicate fan and overtemperature condition
2. The OCP display
3. Circuit breaker(s) tripped

There is no obvious indication for failures 7 – 10 from the power system.

Halt Caused by Power, Fan, or Overtemperature Condition

If a system is stopped because of a power, fan, or overtemperature problem, the console and the OCP should report the problem.

If Power Problem Occurs at Power-Up

If the system has a power problem on a cold start, the motherboard LEDs and the OCP display will indicate a problem. The console, for systems running DIGITAL UNIX or OpenVMS, will also indicate the problem. The console on systems running NT will not print an error message. Causes of power problems are:

- Broken system fan
- Broken CPU fan
- A power supply could be broken and the system could still power up momentarily. (During power-up, an overcurrent condition occurs with two power supplies and is tolerated for a short period but a persistent overcurrent is not.)
- Power control logic on the motherboard could fail
- Interlock failure
- Wire problems
- Temperature problem (unlikely)

Recommended Order for Troubleshooting Failure at Power-Up

1. If the SRM console does not come all the way up, check the console test output on OpenVMS or DIGITAL UNIX systems. Restart the system if the system runs NT and watch for an error message on the OCP display. Replace the FRU indicated.
2. If you can get to the SRM console, use the **show power** command. It will show the last power fault.
3. If neither step one nor step 2 identifies a FRU, replace the motherboard.

3.3 Running Diagnostics — Test Command

The test command runs diagnostics on the entire system, CPU devices, memory devices, and the PCI I/O subsystem. The test command runs only from the SRM console. Ctrl/C stops the test. The console cannot be secure.

Example 3-1 Test Command Syntax

```
P00>>> help test
```

```
FUNCTION
```

```
SYNOPSIS
```

```
test ([-q] [-t <time>] [option])
```

```
where option is:
```

```
cpun
```

```
memn
```

```
pcin
```

```
where n = 0, 1 or * for CPUs and PCIs
```

```
where n = 0 through 7 or * for MEM
```

```
The entire system is tested by default if no is option specified.
```

*NOTE: If you are running the Microsoft Windows NT operating system, switch from AlphaBIOS to the SRM console in order to enter the **test** command. From the AlphaBIOS console, press in the Halt button (the LED will light) and reset the system, or select **DIGITAL UNIX (SRM)** or **OpenVMS (SRM)** from the **Advanced CMOS Setup** screen and reset the system.*

test [-t *time*] [-q] [*option*]

-t *time* Specifies the run time in seconds. The default for system test is 600 seconds (10 minutes).

-q Disables the display of status messages as exerciser processes are started and stopped during testing.

option Either **cpun**, **memn**, or **pcin**, where *n* is 0, 1, or * for CPUs and PCIs; or where *n* is 0 through 7 or * for memory. If nothing is specified, the entire system is tested.

3.4 Releasing Secure Mode

The console cannot be secure for most SRM console commands to run. If the console is not secure, user mode console commands can be entered. See the system manager if the system is secure and you do not know the password.

Example 3-2 Releasing/Reestablishing Secure Mode

```
P00>>> login
Please enter password: xxxx
P00>>>
```

```
[User mode SRM console commands are now available.]
```

```
P00>>> set secure
```

The console command **login** clears secure.

If the password has been forgotten and the system is in secure mode, the procedure for regaining control is:

1. Enter the **login** command P00>>> login
2. At the `please enter password:` prompt, press the Halt button and then press the Return key.

The password is now cleared and the console is in user mode. A new password must be set to put the console into secure mode again.

For a full discussion of securing the console, see your system *User's Guide*.

3.5 Testing an Entire System

A test command with no modifiers runs all exercisers for subsystems and devices on the system. I/O devices tested are supported boot devices. The test runs for 10 minutes.

Example 3-3 Sample Test Command

```
P00>>> test

Console is in diagnostic mode

System test, runtime 600 seconds

Type ^C to stop testing

Configuring system..
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1      DKa500                RRD45  1645
polling ncr1 (NCR 53C810) slot 3, bus 0 PCI, hose 1   SCSI Bus ID 7
dkb200.2.0.3.1      DKb200                RZ29B  0007
dkb400.4.0.3.1      DKb400                RZ29B  0007
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0     DVA0                  RX23
polling tulip0 (DECchip 21040-AA) slot 2, bus 0 PCI, hose 1
ewa0.0.0.2.1: 08-00-2B-E5-B4-1A

Testing EWA0 network device

Testing VGA (alphanumeric mode only)

Starting background memory test, affinity to all CPUs..
Starting processor/cache thrasher on each CPU..
Starting processor/cache thrasher on each CPU..

Testing SCSI disks (read-only)
No CD-ROM present, skipping embedded SCSI test
Testing other SCSI devices (read-only)..
Testing floppy drive (dva0, read-only)
```


ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00003047	memtest	memory	1	0	0	134217728
00003050	memtest	memory	205	0	0	213883392
00003059	memtest	memory	192	0	0	200253568
00003062	memtest	memory	192	0	0	200253568
00003084	memtest	memory	80	0	0	82827392
000030d8	exer_kid	dkb200.2.0.3	26	0	0	0
000030d9	exer_kid	dkb400.4.0.3	26	0	0	0
0000310d	exer_kid	dva0.0.0.100	0	0	0	0
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00003047	memtest	memory	1	0	0	432013312
00003050	memtest	memory	635	0	0	664716032
00003059	memtest	memory	619	0	0	647940864
00003062	memtest	memory	620	0	0	648989312
00003084	memtest	memory	263	0	0	274693376
000030d8	exer_kid	dkb200.2.0.3	90	0	0	0
000030d9	exer_kid	dkb400.4.0.3	90	0	0	0
0000310d	exer_kid	dva0.0.0.100	0	0	0	0
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00003047	memtest	memory	1	0	0	727711744
00003050	memtest	memory	1054	0	0	1104015744
00003059	memtest	memory	1039	0	0	1088289024
00003062	memtest	memory	1041	0	0	1090385920
00003084	memtest	memory	447	0	0	467607808
000030d8	exer_kid	dkb200.2.0.3	155	0	0	0
000030d9	exer_kid	dkb400.4.0.3	155	0	0	0
0000310d	exer_kid	dva0.0.0.100	1	0	0	0

Testing aborted. Shutting down tests.
Please wait..

System test complete

^C
P00>>>

3.5.1 Testing Memory

The test mem command tests individual memory devices or all memory. The test shown in Example 3–4 runs for 2 minutes.

Example 3–4 Sample Test Memory Command

```
P00>>> test memory
Console is in diagnostic mode
System test, runtime 120 seconds
```

Type ^C to stop testing

```
Starting background memory test, affinity to all CPUs..
Starting memory thrasher on each CPU..
Starting memory thrasher on each CPU..
```

ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
000046d7	memtest	memory	1	0 0	48234496	48234496
000046e0	memtest	memory	122	0 0	126862208	126862208
000046e9	memtest	memory	111	0 0	115329280	115329280
000046f2	memtest	memory	109	0 0	113232384	113232384
000046fb	memtest	memory	41	0 0	41937920	41937920
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
000046d7	memtest	memory	1	0 0	226492416	226492416
000046e0	memtest	memory	566	0 0	592373120	592373120
000046e9	memtest	memory	555	0 0	580840192	580840192
000046f2	memtest	memory	554	0 0	579791744	579791744
000046fb	memtest	memory	211	0 0	220174080	220174080
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
000046d7	memtest	memory	1	0 0	404750336	404750336
000046e0	memtest	memory	1011	0 0	1058932480	1058932480
000046e9	memtest	memory	1000	0 0	1047399552	1047399552
000046f2	memtest	memory	999	0 0	1046351104	1046351104
000046fb	memtest	memory	381	0 0	398410240	398410240

ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
000046d7	memtest	memory	1	0	0	583008256
000046e0	memtest	memory	1456	0	0	1525491840
000046e9	memtest	memory	1446	0	0	1515007360
000046f2	memtest	memory	1444	0	0	1512910464
000046fb	memtest	memory	550	0	0	575597952
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
000046d7	memtest	memory	1	0	0	761266176
000046e0	memtest	memory	1901	0	0	1992051200
000046e9	memtest	memory	1892	0	0	1982615168
000046f2	memtest	memory	1889	0	0	1979469824
000046fb	memtest	memory	720	0	0	753834112
ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
000046d7	memtest	memory	1	0	0	937426944
000046e0	memtest	memory	2346	0	0	2458610560
000046e9	memtest	memory	2337	0	0	2449174528
000046f2	memtest	memory	2333	0	0	2444980736
000046fb	memtest	memory	890	0	0	932070272

Memory test complete

Test time has expired...

P00>>>

3.5.2 Testing PCI

The test pci command tests PCI buses and devices. The test runs for 2 minutes.

Example 3-5 Sample Test Command for PCI

```
P00>>> test pci*
Console is in diagnostic mode
System test, runtime 120 seconds

Type ^C to stop testing

Configuring all PCI buses..
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1      DKa500      RRD45  1645
polling ncr1 (NCR 53C810) slot 3, bus 0 PCI, hose 1   SCSI Bus ID 7
dkb200.2.0.3.1      DKb200      RZ29B  0007
dkb400.4.0.3.1      DKb400      RZ29B  0007
polling tulip0 (DECchip 21040-AA) slot 2, bus 0 PCI, hose 1
ewa0.0.0.2.1: 08-00-2B-E5-B4-1A
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0     DVA0      RX23

Testing all PCI buses..

Testing EWA0 network device

Testing VGA (alphanumeric mode only)

Testing SCSI disks (read-only)

Testing floppy (dva0, read-only)
```

ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00002c29	exer_kid	dkb200.2.0.3	27	0 0	0	14642176
00002c2a	exer_kid	dkb400.4.0.3	27	0 0	0	14642176
00002c5e	exer_kid	dva0.0.0.100	0	0 0	0	0

ID	Program	Device	Pass	Hard/Soft	Bytes Written	Bytes Read
00002c29	exer_kid	dkb200.2.0.3	92	0 0	0	48689152
00002c2a	exer_kid	dkb400.4.0.3	92	0 0	0	48689152
00002c5e	exer_kid	dva0.0.0.100	0	0 0	0	286720

Testing aborted. Shutting down tests.
Please wait..

Testing complete

^C
P00>>>

3.6 Other Useful Console Commands

There are several console commands that help diagnose the system.

The **show power** command can be used to identify power, temperature, and fan faults.

Example 3-6 Show Power

```
P00>>> show power
```

	Status
Power Supply 0	good
Power Supply 1	good
System Fans	good
CPU Fans	good
Temperature	good

Current ambient temperature is 20 degrees C
System shutdown temperature is set to 55 degrees C

The system was last reset via a system software reset

0 Environmental events are logged in nvram

The **show memory** command shows memory DIMMs and their starting addresses.

Example 3-7 Show Memory

```
P00>>> show memory
```

Slot	Type	MB	Base
----	----	----	-----
0	DIMM	256	0
1	DIMM	256	20000000
2	DIMM	256	40000000
3	DIMM	256	60000000
Total		1.2GB	

The **show fru** command lists all FRUs in the system.

Example 3-8 Show FRU

P00>>> show fru

Digital Equipment Corporation
AlphaServer 1200

Console V5.0-2 OpenVMS PALcode V1.19-12, Digital UNIX
PALcode V1.21-20

Module Serial #	Part #	Type	Rev	Name	
System Motherboard NI72000047	25147-01	0	0000	mthrbrd0	
Memory 256 MB DIMM	N/A	0	0000	mem0	N/A
Memory 256 MB DIMM	N/A	0	0000	mem1	N/A
Memory 256 MB DIMM	N/A	0	0000	mem2	N/A
Memory 256 MB DIMM	N/A	0	0000	mem3	N/A
CPU (4MB Cache) KA705TRVNS	B3007-AA	3	0000	cpu0	
Bridge (IOD0/IOD1) NI72000047	25147-01	600	0032	iod0/iod1	
PCI Motherboard NI72000047	25147-01	a	0003	saddle0	
Bus 0 iod0 (PCI0)					
Slot Option Name		Type	Rev	Name	
1 PCEB		4828086	0005	pceb0	
2 S3 Trio64/Trio32		88115333	0054	vga0	
3 DECchip 21041-AA		141011	0011	tulip0	
Bus 1 pceb0 (EISA Bridge connected to iod0, slot 1)					
Slot Option Name		Type	Rev	Name	
Bus 0 iod1 (PCI1)					
Slot Option Name		Type	Rev	Name	
1 NCR 53C810		11000	0002	ncr0	
4 QLogic ISP1020		10201077	0005	isp0	

Chapter 4

Error Logs

This chapter provides information on troubleshooting with error logs. The following topics are covered:

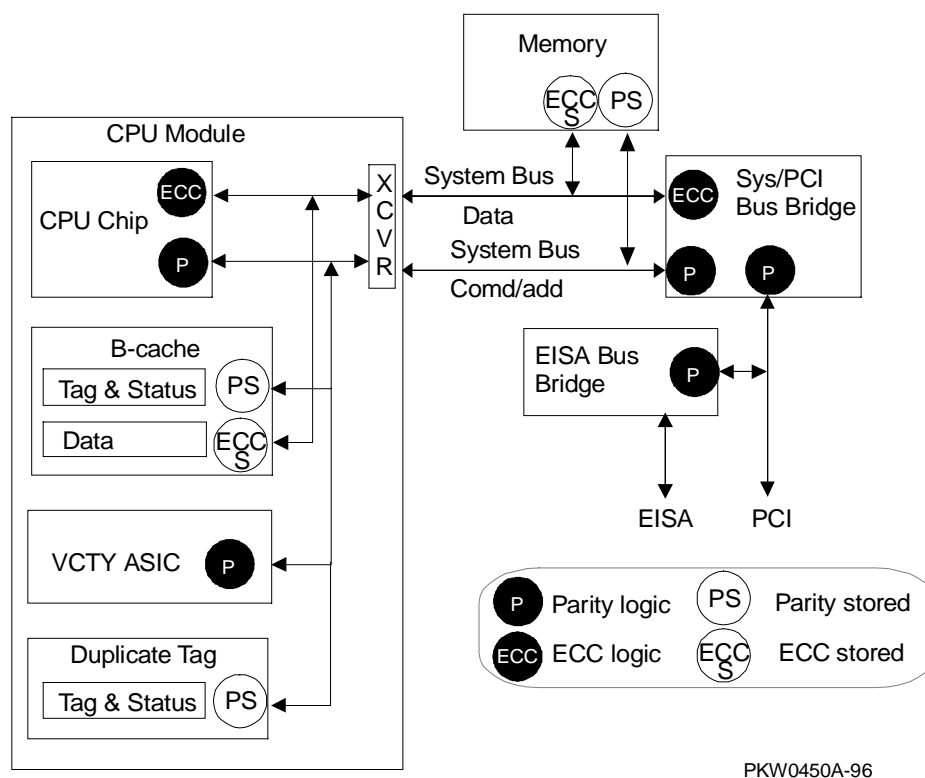
- Using Error Logs
- Using DECEvent
- Error Log Examples and Analysis
- Troubleshooting IOD-Detected Errors
- Double Error Halts and Machine Checks While in PAL Mode

Error registers are described in Chapter 5.

4.1 Using Error Logs

Error detection is performed by CPUs, the IOD, and the EISA to PCI bus bridge. (The IOD is the acronym used by software to refer to the system bus to PCI bus bridge.)

Figure 4-1 Error Detector Placement



Lines Protected	Device
ECC Protected	
System bus data lines	IOD on every transaction, CPU when using the bus
B-cache	IOD on every transaction, CPU when using the bus
Parity Protected	
System bus command/address lines	IOD on every transaction, CPU when using the bus
Duplicate tag store	IOD on every transaction, CPU when using the bus
B-cache index lines	CPU
PCI bus	IOD
EISA bus	EISA bridge

As shown in Figure 4-1 and the accompanying table, the CPU chip is isolated by transceivers (XVER) from the data and command/address lines on the module. This allows the CPU chip access to the duplicate tag and B-cache while the system bus is in use. The CPU detects errors only when it is the consumer of the data. The IOD detects errors on each system bus cycle regardless of whether it is involved in the transaction.

System bus errors detected by the CPU may also be detected by the IOD. It is necessary to check the IOD for errors any time there is a CPU machine check.

- If the CPU sees bad data and the IOD does not, the CPU is at fault.
- If both the CPU and the IOD see bad data on the system bus, either memory or a secondary CPU is the cause. In such a case, the Dirty bit, bit<20>, in the IOD MC_ERR1 Register should be set or clear. If the Dirty bit is set, the source of the data is a CPU's cache destined for a different CPU. If the Dirty bit is not set, memory caused the bad data on the bus. In this case, multiple error log entries occur and must be analyzed together to determine the cause of the error.

4.1.1 Hard Errors

There are two categories of hard errors:

- System-independent errors detected by the CPU. These errors are processor machine checks handled as MCHK 670 interrupts and are:

Internal EV5 or EV56 cache errors
CPU B-cache module errors
- System-dependent errors detected by both the CPU and IOD. These errors are system machine checks handled as MCHK 660 interrupts and are:

CPU-detected external reference errors
IOD hard error interrupts

The IOD can detect hard errors on either side of the bridge.

4.1.2 Soft Errors

There are two categories of soft errors:

- System-independent errors detected and corrected by the CPU. These errors are CPU module correctable errors handled as MCHK 630 interrupts.
- System-dependent errors that are correctable single-bit errors on the system bus and are handled as MCHK 620 interrupts.

4.1.3 Error Log Events

Several different events are logged by OpenVMS and DIGITAL UNIX. Windows NT does not log errors in this fashion.

Table 4-1 Types of Error Log Events

Error Log Event	Description
MCHK 670	Processor machine checks. These are synchronous errors that inform precisely what happened at the time the error occurred. They are detected inside the CPU chip and are fatal errors.
MCHK 660	System machine checks. These are asynchronous errors that are recorded after the error has occurred. Data on exactly what was going on in the machine at the time of the error may not be known. They are fatal errors.
MCHK 630	Processor correctable errors
MCHK 620	System correctable errors
Last fail	Used to collect system bus registers prior to crashing
I/O error interrupt	IOD error interrupts
System environment	Used to provide status on power, fans, and temperature
Configuration	Used to provide system configuration information

4.2 Using DECEvent

DECEvent produces bit-to-text ASCII reports derived from system event entries or user-supplied event logs. The format of the reports is determined by commands, qualifiers, parameters, and keywords appended to the command. The maximum command line length is 255 characters.

DECEvent allows you to do the following:

- Translate event log files into readable reports
- Select alternate input and output files
- Filter input events
- Select alternative reports
- Translate events as they occur
- Maintain and customize your environment with the interactive shell commands

To access on-line help:

OpenVMS

```
$ HELP DIAGNOSE or
$ DIA /INTERACTIVE
DIA> HELP
```

DIGITAL UNIX

```
> man dia or
> dia hlp
```

Privileges necessary to use DECEvent:

- SYSPRV for the utility
- DIAGNOSE to use the /CONTINUOUS qualifier

4.2.1 Translating Event Files

To produce a translated event report using the default event log file, SYS\$ERRORLOG:ERRLOG.SYS, enter the following command:

```
OpenVMS
$ DIAGNOSE
```

```
DIGITAL UNIX
> dia -a
```

The DIAGNOSE command allows DECevent to use built-in defaults. This command produces a full report, directed to the terminal screen, from the input event file, SYS\$ERRORLOG:ERRLOG.SYS. The /TRANSLATE qualifier is understood on the command line.

To select an alternate input file

```
OpenVMS
$ DIAGNOSE ERRORLOG.OLD
```

```
DIGITAL UNIX
> dia -a -f syserr-old.hostname
```

These commands select an alternate input file (ERRORLOG.OLD or syserr-old) as the event log to translate. The file name can contain the directory or path, if needed. Wildcard characters can be used.

To send reports to an output file

```
OpenVMS
$ DIAGNOSE/OUTPUT=ERRLOG_OLD.TXT
```

```
DIGITAL UNIX
> dia -a > syserr-old.txt
```

These commands direct the output of DECevent to ERRLOG_OLD.TXT or syserr -old.txt.

To reverse the order of the input events

```
OpenVMS
$ DIAGNOSE/TRANSLATE/REVERSE
```

```
DIGITAL UNIX
> dia -R
```

These commands reverse the order in which events are displayed. The default order is forward chronologically.

4.2.2 Filtering Events

/INCLUDE and **/EXCLUDE** qualifiers allow you to filter input event log files.

The **/INCLUDE** qualifier is used to create output for devices named in the command.

```
OpenVMS
$ DIAGNOSE/TRANSLATE/INCLUDE=(DISK=RZ,DISK=RA92,CPU)
```

```
DIGITAL UNIX
> dia -i disk=rz disk=ra92 cpu
```

The commands shown here create output using only the entries for RZ disks, RA92 disks, and CPUs.

The **/EXCLUDE** qualifier is used to create output for all devices except those named in the command.

```
OpenVMS
$ DIAGNOSE/TRANSLATE/EXCLUDE=(MEMORY)
```

```
DIGITAL UNIX
> dia -x mem
```

Use the **/BEFORE** and **/SINCE** qualifiers to select events before or after a certain date and time.

OpenVMS

```
$ DIAGNOSE/TRANSLATE/BEFORE=15-JAN-1997:10:30:00
```

or

```
$ DIAGNOSE/TRANSLATE/SINCE=15-JAN-1997:10:30:00
```

DIGITAL UNIX

```
> dia -t s:15-jan-1997 e:20-jan-1997
```

If no time is specified, the default time is 00:00:00, and all events for that day are selected.

The **/BEFORE** and **/SINCE** qualifiers can be combined to select a certain period of time.

OpenVMS

```
$ DIAGNOSE/TRANSLATE/SINCE=15-JAN-1997/BEFORE=20-JAN-1997
```

If no value is supplied with the **/SINCE** or **/BEFORE** qualifiers, DECevent defaults to **TODAY**.

4.2.3 Selecting Alternative Reports

Table 4-2 describes the DECEvent report formats. Report formats are mutually exclusive. No combinations are allowed. The default format is /Full.

Table 4-2 DECEvent Report Formats

Format	Description
/Full	Translates all available information for each event
/Brief	Translates key information for each event
/Terse	Provides binary event information and displays register values and other ASCII messages in a condensed format
/Summary	Produces a statistical summary of the events in the log
/Fsterr	Produces a one-line-per-entry report for disk and tape devices

The syntax is:

```
OpenVMS
$ DIAGNOSE/TRANSLATE/<format>
```

```
DIGITAL UNIX
> dia -o <format>
```

4.3 Error Log Examples and Analysis

The following sections provide examples and analysis of error logs.

4.3.1 MCHK 670 CPU-Detected Failure

The error log in Example 4–1 shows the following:

- ❶ CPU1 logged the error in a system with two CPUs.
- ❷ During a D-ref fill, the External Interface Status Register logged an uncorrectable EEC error. (When a CPU chip does not find data it needs to perform a task in any of its caches, it requests data from off the chip to fill its D-caches. It performs a “D-ref fill.”) Bit<30> is clear, indicating that the source of the error is the B-cache.
- ❸ Neither IOD CAP Error Register saw an error.

The error was detected by a CPU and the data was not on the system bus. Otherwise, the IODs would have seen the error. Therefore, CPU1 is broken.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-1 MCHK 670

```

Logging OS                2. DIGITAL UNIX
System Architecture       2. Alpha
Event sequence number     4.
Timestamp of occurrence   04-APR-1997 17:20:04
Host name                 whip16

System type register      x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)    x00000002 ❶
CPU logging event (mperr) x00000001

Event validity            1. O/S claims event is valid
Event severity            1. Severe Priority
Entry type                100. CPU Machine Check Errors

CPU Minor class           1. Machine check (670 entry)

Software Flags            x0000000300000000
                          IOD 1 Register Subpkt Pres
                          IOD 2 Register Subpkt Pres

Active CPUs               x00000003
Hardware Rev              x00000000
System Serial Number      C1563
Module Serial Number
Module Type                x0000
System Revision            x00000000

* MCHK 670 Regs *
Flags:                    x00000000
PCI Mask                  x0000
Machine Check Reason      x0098
PAL SHADOW REG 0          x00000000
PAL SHADOW REG 1          x00000000
.
.
.
PAL SHADOW REG 6          x00000000
PAL SHADOW REG 7          x00000000
PALTEMP0                  x00000000E87C7A58
PALTEMP1                  xFFFFFFFE8F658000
PALTEMP2                  xFFFFFFC00003C9F40
.
.
.
PALTEMP22                 xFFFFFFC00004F9D60
PALTEMP23                 x00000000E8709A58
Exception Address Reg     xFFFFFFC00003BFB88
                          Native-mode instruction
                          Exception PC x3FFFFFF00000EFEE2

Exception Summary Reg     x00000000
Exception Mask Reg        x00000000
PAL BASE                  x00000000020000
                          Base addr for palcode = x0000000008

Interrupt Summary Reg     x00000000
                          AST requests 3 - 0 x00000000

IBOX Ctrl and Status Reg x000000C160000000
                          Timeout Bit Not Set
                          PAL Shadow Registers Enabled
                          Correctable Err Intrpts Enabled
                          ICACHE BIST Successful
                          TEST_STATUS_H Pin Asserted

```

Icache Par Err Stat Reg	x00000000	
Dcache Par Err Stat Reg	x00000000	
Virtual Address Reg	xFFFFFFFFE8F63BD38	
Memory Mgmt Flt Sts Reg	x00000000166D1	Ref which caused err was a write Ref resulted in DTB miss RA Field x0000000000001B Opcode Field x0000000000002C
Scache Address Reg	xFFFFFFFF0000254BF	
Scache Status Reg	x00000000	
Bcache Tag Address Reg	xFFFFFFFF80E98F7FFF	External cache hit Parity for ds and v bits Cache block dirty Cache block valid Ext cache tag addr parity bit Tag address<38:20> is x000000000000E98
Ext Interface Address Reg	xFFFFFFFF00E984BCF	
Fill Syndrome Reg	x0000000000002B	
Ext Interface Status Reg	xFFFFFFFF104FFFFFF	② Uncorrectable ECC error Error occurred during D-ref fill
LD LOCK	xFFFFFFFF003797340F	
** IOD SUBPACKET -> **		IOD 0 Register Subpacket
WHOAMI	x000000BB	Device ID x0000003B Bcache Size = 2MB VCTY ASIC Rev = 0 Module Revision 0.
Base Address of Bridge	x000000F9E000000	
Dev Type & Rev Register	x06008021	CAP Chip Revision x00000001 Host to PCI Revision x00000003 I/O Backplane Revision x00000003 PCI-EISA Bus Bridge Present on PCI Device Class: Host bus to PCI Bridg
MC-PCI Command Register	x46480FF1	Module Self-Test Passed LED On. Delayed PCI Bus Reads Protocol: Enabled Bridge to PCI Transactions: Enabled Bridge REQUESTS 64 Bit Data Transactions Bridge ACCEPTS 64 Bit Data Transactions PCI Address Parity Check: Enabled MC Bus CMD/Addr Parity Check: Enabled MC Bus NXM Check: Enabled Check ALL Transactions for Errors Use MC_BMSK for 16 Byte Align Blk Mem Wrt Wrt PEND_NUM Threshold: 8. RD_TYPE Memory Prefetch Algorithm: Short RL_TYPE Mem Rd Line Prefetch Type: Medium RM_TYPE Mem Rd Multiple Cmd Type: Long ARB_MODE PCI Arbitration: Round Robin
Memory Host Addr Exten	x00000000	
IO Host Addr Extension	x00000000	
Interrupt Control	x00000003	MC-PCI Intr Enabled Device intr info enabled if en_int=1
Interrupt Request	x00000000	Interrupts asserted x00000000
Interrupt Mask Register 0	x00C50010	
Interrupt Mask Register 1	x00000000	
MC Error Info Register 0	xE0000000	MC bus trans addr <31:4> x0E000000
MC Error Info Register 1	x000E88FD	MC bus trans addr <39:32>x000000FD MC_Command x00000008 Device Id x0000003A
CAP Error Register	x00000000	(no error seen) ③
PCI Bus Trans Error Adr	x00000000	

```

MDPA Status Register      x00000000  MDPA Chip Revision  x00000000
MDPA Error Syndrome Reg   x00000000  Cycle 0 ECC Syndrome x00000000
                                Cycle 1 ECC Syndrome x00000000
                                Cycle 2 ECC Syndrome x00000000
                                Cycle 3 ECC Syndrome x00000000
MDPB Status Register      x00000000  MDPB Chip Revision  x00000000
MDPB Error Syndrome Reg   x00000000  Cycle 0 ECC Syndrome x00000000
                                Cycle 1 ECC Syndrome x00000000
                                Cycle 2 ECC Syndrome x00000000
                                Cycle 3 ECC Syndrome x00000000

** IOD SUBPACKET -> **          IOD 1 Register Subpacket
WHOAMI                    x000000BB  Device ID  x0000003B
                                Bcache Size = 2MB
                                VCTY ASIC Rev = 0
                                Module Revision 0.
Base Address of Bridge     x000000FBE000000
Dev Type & Rev Register    x06008021  CAP Chip Revision  x00000001
                                Host to PCI Revision  x00000003
                                I/O Backplane Revision x00000003
                                PCI-EISA Bus Bridge Present on PCI
                                Device Class: Host bus to PCI Bridg
MC-PCI Command Register   x46480FF1  Module Self-Test Passed LED On.
                                Delayed PCI Bus Reads Protocol: Enabled
                                Bridge to PCI Transactions: Enabled
                                Bridge REQUESTS 64 Bit Data Transactions
                                Bridge ACCEPTS 64 Bit Data Transactions
                                PCI Address Parity Check: Enabled
                                MC Bus CMD/Addr Parity Check: Enabled
                                MC Bus NXM Check: Enabled
                                Check ALL Transactions for Errors
                                Use MC_BMSK for 16 Byte Align Blk Mem Wrt
                                Wrt PEND_NUM Threshold: 8.
                                RD_TYPE Memory Prefetch Algorithm: Short
                                RL_TYPE Mem Rd Line Prefetch Type: Medium
                                RM_TYPE Mem Rd Multiple Cmd Type: Long
                                ARB_MODE PCI Arbitration: Round Robin
Memory Host Addr Exten    x00000000
IO Host Addr Extension    x00000000
Interrupt Control         x00000003  MC-PCI Intr Enabled
                                Device intr info enabled if en_int
                                = 1
Interrupt Request         x00000000  Interrupts asserted x00000000
Interrupt Mask Register 0 x00C50001
Interrupt Mask Register 1 x00000000
MC Error Info Register 0  xE0000000  MC bus trans addr <31:4> x0E000000
MC Error Info Register 1 x000E88FD  MC bus trans addr <39:32> x000000FD
                                MC_Command x00000008
                                Device Id x0000003A

CAP Error Register          x00000000 (no error seen)
PCI Bus Trans Error Adr   xC0018B48
MDPA Status Register      x00000000  MDPA Chip Revision  x00000000
MDPA Error Syndrome Reg   x00000000  Cycle 0 ECC Syndrome x00000000
                                Cycle 1 ECC Syndrome x00000000
                                Cycle 2 ECC Syndrome x00000000
                                Cycle 3 ECC Syndrome x00000000
MDPB Status Register      x00000000  MDPB Chip Revision  x00000000
MDPB Error Syndrome Reg   x00000000  Cycle 0 ECC Syndrome x00000000
                                Cycle 1 ECC Syndrome x00000000
                                Cycle 2 ECC Syndrome x00000000
                                Cycle 3 ECC Syndrome x00000000
PALcode Revision          x00000000  Palcode Rev: 1.21-3

```

③

4.3.2 MCHK 670 CPU and IOD-Detected Failure

The error log in Example 4–2 shows the following:

- ❶ CPU1 logged the error in a system with two CPUs.
- ❷ The External Interface Status Register logged an uncorrectable ECC error during a D-ref fill. (When a CPU chip does not find data it needs to perform a task in any of its caches, it requests data from off the chip to fill its D-cache. It performs a “D-ref fill.”) Bit <30> is set, indicating that the source of the error is memory or the system. Bits <32> and <35> are set, indicating an uncorrectable ECC error and a second external interface hard error, respectively.
- ❸ Both IOD CAP Error Registers logged an error.
- ❹ The command at the time of the error was a read.
- ❺ The bus master at the time of the error was CPU1.
- ❻ The Dirty bit, bit <20> in the MC_ERR1 Register is clear, indicating the data is clean and comes from memory.

The error was detected by a CPU, and the data was on the system bus and is clean. Therefore, a memory module provided the wrong data. (If the Dirty bit had been set, the data would have come from the cache of another CPU.) To determine which memory, see Section 4.4.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-2 MCHK 670 CPU and IOD-Detected Failure

```

Logging OS                2. DIGITAL UNIX
System Architecture       2. Alpha
Event sequence number     6.
Timestamp of occurrence   08-APR-1997 11:27:55
Host name                 whip16

System type register      x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)    x00000002
CPU logging event (mperr) x00000001

Event validity            1. O/S claims event is valid
Event severity            1. Severe Priority
Entry type                100. CPU Machine Check Errors

CPU Minor class           1. Machine check (670 entry)

Software Flags            x00000003
                          IOD 1 Register Subpkt Pres
                          IOD 2 Register Subpkt Pres

Active CPUs               x00000002
Hardware Rev              x00000000
System Serial Number      C1563
Module Serial Number
Module Type               x0000
System Revision           x00000000

* MCHK 670 Regs *
Flags:                    x00000000
PCI Mask                  x0000
Machine Check Reason      x0098
PAL SHADOW REG 0          x00000000
PAL SHADOW REG 1          x00000000
.
.
.
PAL SHADOW REG 6          x00000000
PAL SHADOW REG 7          x00000000
PALTEMP0                  x00000001401A7A90
PALTEMP1                  x00000000000021
.
.
.
PALTEMP23                 x00000000ECE77A58
Exception Address Reg     x000000012005A8B4
                          Native-mode instruction
                          Exception PC x0000000048016A2D

Exception Summary Reg     x00000000
Exception Mask Reg        x00000000
PAL BASE                  x0000000020000
                          Base addr for palcode = x0000000008

Interrupt Summary Reg     x00000000
                          AST requests 3 - 0 x00000000

IBOX Ctrl and Status Reg x000000C164000000
                          Timeout Bit Not Set
                          Floating Point Instr. may be issued
                          PAL Shadow Registers Enabled
                          Correctable Err Intrpts Enabled
                          ICACHE BIST Successful
                          TEST_STATUS_H Pin Asserted

Icache Par Err Stat Reg  x00000000
Dcache Par Err Stat Reg  x00000000

```

```

Virtual Address Reg      x00000001407D6000
Memory Mgmt Flt Sts Reg x00000000011A10
                          Ref resulted in DTB miss
                          RA Field x0000000008

                          Opcode Field x00000000000023
Scache Address Reg      xFFFFFFF00000254BF
Scache Status Reg       x00000000
Bcache Tag Address Reg  xFFFFFFF80286F7FFF
                          External cache hit
                          Parity for ds and v bits
                          Cache block dirty
                          Cache block valid
                          Ext cache tag addr parity bit
                          Tag address<38:20> is x000000000000286
Ext Interface Address Reg xFFFFFFF0028681A8F
Fill Syndrome Reg       x00000000004B00
Ext Interface Status Reg xFFFFFFF984FFFFFF ②
                          Uncorrectable ECC error
                          Error occurred during D-ref fill
                          Second external interface hard
                          error
LD LOCK                  xFFFFFFF000020040F

** IOD SUBPACKET -> **
WHOAMI                   x0000000BF
                          IOD 0 Register Subpacket
                          Device ID x0000003F
                          Bcache Size = 2MB
                          VCTY ASIC Rev = 0
                          Module Revision 0.
Base Address of Bridge   x0000000F9E000000
Dev Type & Rev Register  x06008021
                          CAP Chip Revision x00000001
                          Host to PCI Revision x00000003
                          I/O Backplane Revision x00000003
                          PCI-EISA Bus Bridge Present on PCI
                          Device Class: Host bus to PCI Bridg
MC-PCI Command Register x46480FF1
                          Module Self-Test Passed LED On.
                          Delayed PCI Bus Reads Protocol: Enabled
                          Bridge to PCI Transactions: Enabled
                          Bridge REQUESTS 64 Bit Data Transactions
                          Bridge ACCEPTS 64 Bit Data Transactions
                          PCI Address Parity Check: Enabled
                          MC Bus CMD/Addr Parity Check: Enabled
                          MC Bus NXM Check: Enabled
                          Check ALL Transactions for Errors
                          Use MC_BMSK for 16 Byte Align Blk Mem Wrt
                          Wrt PEND_NUM Threshold: 8.
                          RD_TYPE Memory Prefetch Algorithm: Short
                          RL_TYPE Mem Rd Line Prefetch Type: Medium
                          RM_TYPE Mem Rd Multiple Cmd Type: Long
                          ARB_MODE PCI Arbitration: Round Robin
Memory Host Addr Exten  x00000000
IO Host Addr Extension  x00000000
Interrupt Control        x00000003
                          MC-PCI Intr Enabled
                          Device intr info enabled if en_int
                          = 1
Interrupt Request        x00810000
                          Interrupts asserted x00010000
                          Hard Error
Interrupt Mask Register 0 x00C50010
Interrupt Mask Register 1 x00000000
MC Error Info Register 0 x28681A80
                          MC bus trans addr <31:4> x028681A8 ⑥
MC Error Info Register 1 x800ED800
                          MC bus trans addr <39:32> x00000000
                          MC_Command x00000018 ④
                          Device Id x0000003B ⑤

```



```

MC error info valid
CAP Error Register      xC0000000  Uncorrectable ECC err det by MDPB
                          MC error info latched ③

PCI Bus Trans Error Adr x000003FD
MDPA Status Register    x00000000  MDPB Chip Revision  x00000000
MDPA Error Syndrome Reg x00000000  Cycle 0 ECC Syndrome x00000000
                          Cycle 1 ECC Syndrome x00000000
                          Cycle 2 ECC Syndrome x00000000
                          Cycle 3 ECC Syndrome x00000000
MDPB Status Register    x00000000  MDPB Chip Revision  x00000000
                          MDPB Error Syndrome of
                          uncorrectable read error

MDPB Error Syndrome Reg x00000000  Cycle 0 ECC Syndrome x00000000
                          Cycle 1 ECC Syndrome x00000000
                          Cycle 2 ECC Syndrome x00000000
                          Cycle 3 ECC Syndrome x00000000

** IOD SUBPACKET -> **
WHOAMI                  x000000BF  IOD 1 Register Subpacket
                          Device ID  x0000003F
                          Bcache Size = 2MB
                          VCTY ASIC Rev = 0
                          Module Revision 0.

Base Address of Bridge   x000000FBE000000
Dev Type & Rev Register  x06008021  CAP Chip Revision  x00000001
                          Host to PCI Revision  x00000003
                          I/O Backplane Revision x00000003
                          PCI-EISA Bus Bridge Present on PCI
                          Device Class: Host bus to PCI Bridg
MC-PCI Command Register x46480FF1  Module Self-Test Passed LED On.
                          Delayed PCI Bus Reads Protocol: Enabled
                          Bridge to PCI Transactions: Enabled
                          Bridge REQUESTS 64 Bit Data Transactions
                          Bridge ACCEPTS 64 Bit Data Transactions
                          PCI Address Parity Check: Enabled
                          MC Bus CMD/Addr Parity Check: Enabled
                          MC Bus NXM Check: Enabled
                          Check ALL Transactions for Errors
                          Use MC_BMSK for 16 Byte Align Blk Mem Wrt
                          Wrt PEND_NUM Threshold: 8.
                          RD_TYPE Memory Prefetch Algorithm: Short
                          RL_TYPE Mem Rd Line Prefetch Type: Medium
                          RM_TYPE Mem Rd Multiple Cmd Type: Long
                          ARB_MODE PCI Arbitration: Round Robin

Memory Host Addr Exten  x00000000
IO Host Addr Extension  x00000000
Interrupt Control        x00000003  MC-PCI Intr Enabled
                          Device intr info enabled if en_int
                          = 1

Interrupt Request        x00800000  Interrupts asserted x00000000
                          Hard Error

Interrupt Mask Register 0 x00C50001
Interrupt Mask Register 1 x00000000

MC Error Info Register 0 x28681A80  MC bus trans addr <31:4> x028681A8 ⑥
MC Error Info Register 1 x800FD800  MC bus trans addr <39:32> x00000000
                          MC_Command x00000018 ④
                          Device Id x0000003B ⑤
                          MC error info valid
CAP Error Register      xC0000000  Uncorrectable ECC err det by MDPB
                          MC error info latched ③

PCI Bus Trans Error Adr x00000000
MDPA Status Register    x00000000  MDPB Chip Revision  x00000000
MDPA Error Syndrome Reg x00000000  Cycle 0 ECC Syndrome x00000000

```

		Cycle 1 ECC Syndrome x00000000
		Cycle 2 ECC Syndrome x00000000
		Cycle 3 ECC Syndrome x00000000
MDPB Status Register	x00000000	MDPB Chip Revision x00000000
		MDPB Error Syndrome of
		uncorrectable read error
MDPB Error Syndrome Reg	x00000000	Cycle 0 ECC Syndrome x00000000
		Cycle 1 ECC Syndrome x00000000
		Cycle 2 ECC Syndrome x00000000
		Cycle 3 ECC Syndrome x00000000
PALcode Revision		Palcode Rev: 1.21-3

4.3.3 MCHK 670 Read Dirty CPU-Detected Failure

The error log in Example 4–3 shows the following:

- ❶ CPU0 logged the error in a system with two CPUs.
- ❷ The External Interface Status Register records an uncorrectable ECC error from the system (bit <30> set).
- ❸ Both IOD CAP Error Registers logged an error.
- ❹ The MC Error Info Registers 0 and 1 have captured the error information.
- ❺ The commander at the time of the error was CPU0 (known from MC_ERR1).
- ❻ The command on the bus at the time was a read memory command.
- ❼ The address read was a memory address, not an I/O address.
- ❽ The data associated with the read was dirty.

From this information you know CPU0 requested data that was dirty; therefore, memory did not provide it, nor did an I/O device. Only another CPU could have provided the data from its cache. There is only one other CPU in this system, and it is faulty. See Section 4.4 for a procedure designed to help with IOD-detected errors.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-3 MCHK 670 Read Dirty Failure

```

Logging OS                2. DIGITAL UNIX
System Architecture       2. Alpha
Event sequence number     4.
Timestamp of occurrence   08-APR-1997 10:20:37
Host name                 sect06

System type register      x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)   x00000002
CPU logging event (mperr) x00000000 ❶

Event validity            1. O/S claims event is valid
Event severity            1. Severe Priority
Entry type                100. CPU Machine Check Errors

CPU Minor class           1. Machine check (670 entry)

Software Flags            x0000000300000000
                           IOD 0 Register Subpkt Pres
                           IOD 1 Register Subpkt Pres

Active CPUs               x00000003
Hardware Rev              x00000000
System Serial Number      C1563
Module Serial Number
Module Type                x0000
System Revision            x00000000

* MCHK 670 Regs *
Flags:                    x00000000
PCI Mask                   x0000
Machine Check Reason       x0098 Fatal Alpha Chip Detected HardError
PAL SHADOW REG 0           x0000000000000000
PAL SHADOW REG 1           x0000000000000000
PAL SHADOW REG 2           x0000000000000000
PAL SHADOW REG 3           x0000000000000000
PAL SHADOW REG 4           x0000000000000000
PAL SHADOW REG 5           x0000000000000000
PAL SHADOW REG 6           x0000000000000000
PAL SHADOW REG 7           x0000000000000000
PALTEMP0                   xFFFFFFC00006C00C0
PALTEMP1                   x00000000000061A8
PALTEMP2                   xFFFFFFC00004E1E00
.
.
.
PALTEMP22                  xFFFFFFC00006530E0
PALTEMP23                  x00000000003D2BA58
Exception Address Reg       xFFFFFFC000047395C
                           Native-mode Instruction
                           Exception PC x3FFFFFF000011CE57

Exception Summary Reg      x0000000000000000
Exception Mask Reg         x0000000000000000
PAL Base Address Reg       x00000000000020000
                           Bse Addr for PALcode: = x0000000000000008

Interrupt Summary Reg      x0000000000200000
                           External HW Interrupt at IPL21
                           AST Requests 3-0: x0000000000000000

IBOX Ctrl and Status Reg   x0000000C16000000
                           Timeout Counter Bit Clear.
                           IBOX Timeout Counter Enabled.

```

```

Floating Point Instructions will
cause FEN Exceptions.
PAL Shadow Registers Enabled.
Correctable Error Interrupts
Enabled.
ICACHE BIST (Self Test) Was
Successful.
TEST_STATUS_H Pin Asserted
Icache Par Err Stat Reg  x0000000000000000
Dcache Par Err Stat Reg  x0000000000000000
Virtual Address Reg       x0000000000004400
Memory Mgmt Flt Sts Reg  x00000000000005D10
                           If Err, Reference Resulted in DTB
                           Miss
                           Fault Inst RA Field:  x00000000000000014
                           Fault Inst Opcode:  x000000000000000B
Scache Address Reg        xFFFFFF00000254BF
Scache Status Reg         x0000000000000000
Bcache Tag Address Reg    xFFFFFF8007EE2FFF
                           Last Bcache Access Resulted in a
                           Miss.
                           Value of Parity Bit for Tag Control
                           Status
                           Bits Dirty, Shared & Valid is Set.
                           Value of Tag Control Dirty Bit is
                           Clear.
                           Value of Tag Control Shared Bit is
                           Clear.
                           Value of Tag Control Valid Bit is
                           Clear.
                           Value of Parity Bit Covering Tag
                           Store address Bits is Set.
                           Tag Address<38:20> Is:  x0000000000000007E
Ext Interface Address Reg  xFFFFFF0007FBF08F
Fill Syndrome Reg         x0000000000000D189
Ext Interface Status Reg  xFFFFFFF944FFFFFF
                           2
                           Error Source is Memory or System
                           UNCORRECTABLE ECC ERROR
                           Error Occurred During D-ref Fill
                           Error
LD LOCK                    xFFFFFF0007FBF00F
** IOD SUBPACKET -> **    IOD 0 Register Subpacket
WHOAMI                    x000000BA  Module Revision  0.
                           VCTY ASIC Rev = 0
                           Bcache Size = 2MB
                           MID  2.
                           GID  7.
Base Address of Bridge    x000000F9E0000000
Dev Type & Rev Register    x06008021  CAP Chip Revision  x00000001
                           Host to PCI Revision  x00000003
                           I/O Backplane Revision x00000003
                           PCI-EISA Bus Bridge Present on PCI
                           Device Class: Host bus to PCI Bridg
MC-PCI Command Register   x06480FF1  Module SelfTest Passed LED on
                           Delayed PCI Bus Reads Protocol:  Enabled
                           Bridge to PCI Transactions: Enabled
                           Bridge REQUESTS 64 Bit Data Transactions
                           Bridge ACCEPTS 64 Bit Data Transactions
                           PCI Address Parity Check: Enabled
                           MC Bus CMD/Addr Parity Check: Enabled
                           MC Bus NXM Check: Enabled

```

```

Check ALL Transactions for Errors
Use MC_BMSK for 16 Byte Align Blk Mem Wrt
Wrt PEND_NUM Threshold: 8.
RD_TYPE Memory Prefetch Algorithm: Short
RL_TYPE Mem Rd Line Prefetch Type: Medium
RM_TYPE Mem Rd Multiple Cmd Type: Long
ARB_MODE Arbitration: MC-PCI Priority Mode
Mem Host Address Ext Reg x00000000 HAE Sparse Mem Adr<31:27> x00000000
IO Host Adr Ext Register x00000000 PCI Upper Adr Bits<31:25> x00000000
Interrupt Ctrl Register x00000003 Write Device Interrupt Info Struct:Enabled
Interrupt Request x00800000 Interrupts asserted x00000000
Hard Error
Interrupt Mask0 Register x00C50010
Interrupt Mask1 Register x00000000
MC Error Info Register 0 x07FBF080
MC Bus Trans Adr<31:4>: 7FBF080
MC Error Info Register 1 x801E8800 MC bus trans addr <39:32> x00000000 ⑦
MC Command is Read0-Mem ⑥
Device ID 2 x00000002 ⑤
MC bus error assoc w read/dirty ⑧
MC error info valid
CAP Error Register xE0000000 Uncorrectable ECC err det by MDPA ③
Uncorrectable ECC err det by MDPB
MC error info latched ④
Sys Environmental Regs x00000000
PCI Bus Trans Error Adr x00000000
MDPA Status Register x00000000 MDPA Status Register Data Not Valid
MDPA Error Syndrome Reg x00000000 MDPA Syndrome Register Data Not Valid
MDPB Status Register x00000000 MDPB Status Register Data Not Valid
MDPB Error Syndrome Reg x000D00D0 MDPB Syndrome Register Data Not Valid
** IOD SUBPACKET -> ** IOD 1 Register Subpacket
WHOAMI x000000BA Module Revision 0.
VCTY ASIC Rev = 0
Bcache Size = 2MB
MID 2.
GID 7.
Base Address of Bridge x000000FBE0000000
Dev Type & Rev Register x06008021 CAP Chip Revision x00000001
Host to PCI Revision x00000003
I/O Backplane Revision x00000003
PCI-EISA Bus Bridge Present on PCI
Device Class: Host bus to PCI Bridg
MC-PCI Command Register x06480FF1 Module SelfTest Passed LED on
Delayed PCI Bus Reads Protocol: Enabled
Bridge to PCI Transactions: Enabled
Bridge REQUESTS 64 Bit Data Transactions
Bridge ACCEPTS 64 Bit Data Transactions
PCI Address Parity Check: Enabled
MC Bus CMD/Addr Parity Check: Enabled
MC Bus NXM Check: Enabled
Check ALL Transactions for Errors
Use MC_BMSK for 16 Byte Align Blk Mem Wrt
Wrt PEND_NUM Threshold: 8.
RD_TYPE Memory Prefetch Algorithm: Short
RL_TYPE Mem Rd Line Prefetch Type: Medium
RM_TYPE Mem Rd Multiple Cmd Type: Long
ARB_MODE Arbitration: MC-PCI Priority Mode
Mem Host Address Ext Reg x00000000 HAE Sparse Mem Adr<31:27> x00000000
IO Host Adr Ext Register x00000000 PCI Upper Adr Bits<31:25> x00000000

```

Interrupt Ctrl Register	x00000003	Write Device Interrupt Info Struct:Enabled
Interrupt Request	x00800001	Interrupts asserted x00000001
		Hard Error
Interrupt Mask0 Register	x00C50001	
Interrupt Mask1 Register	x00000000	
MC Error Info Register 0	x07FBF080	
		MC Bus Trans Addr<31:4>: 7FBF080 ⑦
MC Error Info Register 1	x801E8800	MC bus trans addr <39:32> x00000000
		MC Command is Read0-Mem ⑥
		Device ID 2 x00000002 ⑤
		MC bus error assoc w read/dirty ⑧
		MC error info valid
CAP Error Register	xE0000000	Uncorrectable ECC err det by MDPB ③
Uncorrectable ECC err det by MDPB		
		MC error info latched ④
Sys Environmental Regs	x00000000	
PCI Bus Trans Error Adr	x00000000	
MDPA Status Register	x00000000	MDPA Status Register Data Not Valid
MDPA Error Syndrome Reg	x00000000	MDPA Syndrome Register Data Not Valid
MDPB Status Register	x00000000	MDPB Status Register Data Not Valid
MDPB Error Syndrome Reg	x000D00D0	MDPB Syndrome Register Data Not Valid
PALcode Revision		Palcode Rev: 1.21-3

4.3.4 MCHK 660 IOD-Detected Failure (System Bus Error)

The error log in Example 4-4 shows the following:

- ❶ CPU0 logged the error in a system with two CPUs.
- ❷ The External Interface Status Register does not record an error.
- ❸ Both IOD CAP Error Registers logged an error.
- ❹ The MC Error Info Registers 0 and 1 captured the error information.
- ❺ The commander at the time of the error was CPU1 (known from MC_ERR1).
- ❻ The command on the bus at the time was a write-back memory command.

Since this is an MCHK 660, the IOD detected the error on the bus, and CPU0 is logging the error. CPU0 registers are not important in this case since it is servicing the IOD interrupt. There are three devices that can put data on the system bus: CPUs, memory, or an IOD. From MC_ERR Register 1 we know that at the time of the error CPU1 put bad data on the bus while writing to memory. See Section 4.4 for a procedure designed to help with IOD-detected errors.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-4 MCHK 660 IOD-Detected Failure (System Bus Error)

Logging OS	2. DIGITAL UNIX
System Architecture	2. Alpha
Event sequence number	6.
Timestamp of occurrence	04-APR-1996 17:20:04
Host name	whipl6
System type register	x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)	x00000002
CPU logging event (mperr)	x00000000 ❶
Event validity	1. O/S claims event is valid
Event severity	1. Severe Priority
Entry type	100. CPU Machine Check Errors
CPU Minor class	2. 660 Entry
Software Flags	x0000000300000000 IOD 1 Register Subpkt Pres IOD 2 Register Subpkt Pres
Active CPUs	x00000003
Hardware Rev	x00000000
System Serial Number	C1563
Module Serial Number	
Module Type	x0000
System Revision	x00000000
* MCHK 660 Regs *	
Flags:	x00000000
PCI Mask	x0000
Machine Check Reason	x0202
PAL SHADOW REG 0	x00000000
.	
.	
PAL SHADOW REG 7	x00000000
PALTEMP0	x0000000007
.	
.	
PALTEMP23	x00000000047FDA58
Exception Address Reg	xFFFFFC000038D784 Native-mode instruction Exception PC x3FFFFFF00000E35E1
Exception Summary Reg	x00000000
Exception Mask Reg	x00000000
PAL BASE	x00000000020000 Base addr for palcode = x0000000008
Interrupt Summary Reg	x00000000200000 EXT. HW interrupt at IPL21 AST requests 3 - 0 x00000000
IBOX Ctrl and Status Reg	x0000000C16000000 Timeout Bit Not Set PAL Shadow Registers Enabled Correctable Err Intrpts Enabled ICACHE BIST Successful TEST_STATUS_H Pin Asserted
Icache Par Err Stat Reg	x00000000
Dcache Par Err Stat Reg	x00000000
Virtual Address Reg	FFFFFFFFFFFF800130
Memory Mgmt Flt Sts Reg	x00000000014990 Ref resulted in DTB miss RA Field x0000000006

```

                                Opcode Field    x000000000000029
Scache Address Reg    xFFFFFFF0000024EAF
Scache Status Reg     x00000000
Bcache Tag Address Reg xFFFFFFF80FFED6FFF
                                Parity for ds and v bits
                                Cache block dirty
                                Cache block valid
                                Tag address<38:20> is    x000000000000FFE
Ext Interface Address Reg xFFFFFFF00FC00000F
Fill Syndrome Reg     x000000000000C5D2
Ext Interface Status Reg xFFFFFFF004FFFFFFF
                                Error occurred during D-ref fill ②
LD LOCK               xFFFFFFF000020065F

** IOD SUBPACKET -> **
WHOAMI                x000000BA  IOD 0 Register Subpacket
                                Device ID x0000003A
                                Bcache Size = 2MB
                                VCTY ASIC Rev = 0
                                Module Revision 0.
Base Address of Bridge x000000F9E0000000
Dev Type & Rev Register x06008021  CAP Chip Revision    x00000001
                                Host to PCI Revision    x00000003
                                I/O Backplane Revision  x00000003
                                PCI-EISA Bus Bridge Present on PCI
                                Device Class: Host bus to PCI Bridg
MC-PCI Command Register x46480FF1  Module Self-Test Passed LED On.
                                Delayed PCI Bus Reads Protocol: Enabled
                                Bridge to PCI Transactions: Enabled
                                Bridge REQUESTS 64 Bit Data Transactions
                                Bridge ACCEPTS 64 Bit Data Transactions
                                PCI Address Parity Check: Enabled
                                MC Bus CMD/Addr Parity Check: Enabled
                                MC Bus NXM Check: Enabled
                                Check ALL Transactions for Errors
                                Use MC_BMSK for 16 Byte Align Blk Mem Wrt
                                Wrt PEND_NUM Threshold: 8.
                                RD_TYPE Memory Prefetch Algorithm: Short
                                RL_TYPE Mem Rd Line Prefetch Type: Medium
                                RM_TYPE Mem Rd Multiple Cmd Type: Long
                                ARB_MODE PCI Arbitration: Round Robin
Memory Host Addr Exten x00000000
IO Host Addr Extension x00000000
Interrupt Control      x00000003  MC-PCI Intr Enabled
                                Device intr info enabled if en_int
                                = 1
Interrupt Request      x00800000  Interrupts asserted x00000000
                                Hard Error
Interrupt Mask Register 0 x00C50010
Interrupt Mask Register 1 x00000000
MC Error Info Register 0 x4A26DBF0  MC bus trans addr <31:4> x04A26DBF
MC Error Info Register 1 x800ED600  MC bus trans addr <39:32> x00000000
                                MC_Command x00000016 ⑥
                                Device Id x0000003B ⑤
                                MC error info valid
CAP Error Register     xA0000000  Uncorrectable ECC err det by MDPA ③
                                MC error info latched ④
PCI Bus Trans Error Adr x00000000
MDPA Status Register   x80000000  MDPA Chip Revision x00000000
                                MDPA Error Syndrome of
                                uncorrectable read error
MDPA Error Syndrome Reg x1E00001E  Cycle 0 ECC Syndrome x000000000000001E
                                Cycle 1 ECC Syndrome x00000000
                                Cycle 2 ECC Syndrome x00000000

```

MDPB Status Register	x00000000	Cycle 3 ECC Syndrome x00000000000001E
MDPB Error Syndrome Reg	x00000000	MDPB Chip Revision x00000000
		Cycle 0 ECC Syndrome x00000000
		Cycle 1 ECC Syndrome x00000000
		Cycle 2 ECC Syndrome x00000000
		Cycle 3 ECC Syndrome x00000000
 ** IOD SUBPACKET -> **		
WHOAMI	x000000BA	IOD 1 Register Subpacket
		Device ID x0000003A
		Bcache Size = 2MB
		VCTY ASIC Rev = 0
		Module Revision 0.
Base Address of Bridge	x000000FBE0000000	
Dev Type & Rev Register	x06008021	CAP Chip Revision x00000001
		Host to PCI Revision x00000003
		I/O Backplane Revision x00000003
		PCI-EISA Bus Bridge Present on PCI
		Device Class: Host bus to PCI Bridg
MC-PCI Command Register	x46480FF1	Module Self-Test Passed LED On.
		Delayed PCI Bus Reads Protocol: Enabled
		Bridge to PCI Transactions: Enabled
		Bridge REQUESTS 64 Bit Data Transactions
		Bridge ACCEPTS 64 Bit Data Transactions
		PCI Address Parity Check: Enabled
		MC Bus CMD/Addr Parity Check: Enabled
		MC Bus NXM Check: Enabled
		Check ALL Transactions for Errors
		Use MC_BMSK for 16 Byte Align Blk Mem Wrt
		Wrt PEND_NUM Threshold: 8.
		RD_TYPE Memory Prefetch Algorithm: Short
		RL_TYPE Mem Rd Line Prefetch Type: Medium
		RM_TYPE Mem Rd Multiple Cmd Type: Long
		ARB_MODE PCI Arbitration: Round Robin
Memory Host Addr Exten	x00000000	
IO Host Addr Extension	x00000000	
Interrupt Control	x00000003	MC-PCI Intr Enabled
		Device intr info enabled if en_int = 1
Interrupt Request	x00800000	Interrupts asserted x00000000
		Hard Error
Interrupt Mask Register 0	x00C50001	
Interrupt Mask Register 1	x00000000	
MC Error Info Register 0	x4A26DBF0	MC bus trans addr <31:4> x04A26DBF
MC Error Info Register 1	x800ED600	MC bus trans addr <39:32> x00000000
		MC_Command x00000016 ⑥
		Device Id x0000003B ⑤
		MC error info valid
CAP Error Register	xA0000000	Uncorrectable ECC err det by MDPA ③
		MC error info latched ④
PCI Bus Trans Error Adr	x00000000	
MDPA Status Register	x80000000	MDPA Chip Revision x00000000
		MDPA Error Syndrome of uncorrectable read error
MDPA Error Syndrome Reg	x1E00001E	Cycle 0 ECC Syndrome x00000000
		Cycle 1 ECC Syndrome x00000000
		Cycle 2 ECC Syndrome x00000000
		Cycle 3 ECC Syndrome x00000000
MDPB Status Register	x00000000	MDPB Chip Revision x00000000
MDPB Error Syndrome Reg	x00000000	Cycle 0 ECC Syndrome x00000000
		Cycle 1 ECC Syndrome x00000000
		Cycle 2 ECC Syndrome x00000000
		Cycle 3 ECC Syndrome x00000000
PALcode Revision		Palcode Rev: 1.21-3

4.3.5 MCHK 660 IOD-Detected Failure (PCI Error)

The error log in Example 4–5 shows the following:

- ❶ CPU 0 logged the error in a system with two CPUs.
- ❷ The MCHK 660 register gives the reason for the machine check as an IOD-detected hard error or a Dtag Parity Error (if cached CPU)
- ❸ The External Interface Status Register records that the error occurred during a D-ref Fill but does not indicate what the error is.
- ❹ The CAP Error Register for IOD0 did not see an error.
- ❺ The CAP Error Register for IOD1, however, records a serious PCI error.
- ❻ The MC Error Info Registers 0 and 1 are not valid since the valid bit, <31> is not set. Exactly what was happening at the time of the error is not known.
- ❼ There is a PCI Subpacket from PCI1 with four nodes on it. Two devices on the PCI bus did not see an error, however two did, the Mylex DAC960 and the DEC_KZPSA. Either device could have caused the parity error.

Since this is an MCHK 660, the IOD detected the error on the bus, and CPU0 is logging the error. CPU0 registers are not important in this case since it is servicing the IOD interrupt. There are three devices that can put data on the system bus: CPUs, memory, or an IOD. The CAP Error Register for IOD1 saw a serious error and the MC Error Info Register was not able to capture error information. The presence of PCI Subpackets informs the diagnosis summarized by ❷.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-5 MCHK 660 IOD-Detected Failure (PCI Error)

```

Timestamp of occurrence      19-AUG-1997 12:53:41
Host name                    sect04

System type register         x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)      x00000002
CPU logging event (mperr)   x00000000 ❶
Event validity               1. O/S claims event is valid
Event severity               1. Severe Priority
Entry type                   100. CPU Machine Check Errors

CPU Minor class              2. 660 Entry

Software Flags               x0000002300000000
                             IOD 0 Register Subpkt Pres
                             IOD 1 Register Subpkt Pres
                             PCI 1 Bus Snapshot Present

Active CPUs                  x00000003
Hardware Rev                 x00000000
System Serial Number         GA12000000
Module Serial Number
Module Type                  x0000
System Revision              x00000000

* MCHK 660 Regs * ❷
Flags:                       x00000000
PCI Mask                     x0002
Machine Check Reason         x0202 IOD-Detected Hard Error -OR-
                             DTag Parity Error (If Cached CPU)

PAL SHADOW REG 0             x0000000000000000
PAL SHADOW REG 1             x0000000000000000
PAL SHADOW REG 2             x0000000000000000
PAL SHADOW REG 3             x0000000000000000
PAL SHADOW REG 4             x00000B6D00000000
PAL SHADOW REG 5             x0000000000000000
PAL SHADOW REG 6             x0000000000000000
PAL SHADOW REG 7             x0000000000000000
PALTEMP0                     x000000000000000B6
PALTEMP1                     x0000000000000001
PALTEMP2                     xFFFFFFC00003B8B90
.
.
.
PALTEMP22                    xFFFFFFC000052E3A0
PALTEMP23                    x0000000002729A38
Exception Address Reg        x00000001200077F0
                             Native-mode Instruction
                             Exception PC x0000000048001DFC

Exception Summary Reg        x0000000000000000
Exception Mask Reg           x0000000000000000
PAL Base Address Reg         x000000000014000
                             Base Addr for PALcode: x0000000000000005

Interrupt Summary Reg        x0000000000020000
                             External HW Interrupt at IPL21
                             AST Requests 3-0: x0000000000000000

IBOX Ctrl and Status Reg     x000000C160020000
                             Timeout Counter Bit Clear.
                             IBOX Timeout Counter Enabled.

```

```

Floating Point Instructions will Cause
FEN Exceptions.
PAL Shadow Registers Enabled.
Correctable Error Interrupts Enabled.
ICACHE BIST (Self Test) Was Successful.
TEST_STATUS_H Pin Asserted

Icache Par Err Stat Reg  x0000000000000000
Dcache Par Err Stat Reg  x0000000000000000
Virtual Address Reg       x00000000140008000
Memory Mgmt Flt Sts Reg  x00000000000005F10

If Err, Reference Resulted in DTB Miss
Fault Inst RA Field:  x0000000000000001C

Fault Inst Opcode:  x000000000000000B
Scache Address Reg      xFFFFFFF0000018FEF
Scache Status Reg       x0000000000000000
Bcache Tag Address Reg  xFFFFFFF8061CD0FFF

Last Bcache Access Resulted in a Miss.
Value of Parity Bit for Tag Control Status
Bits Dirty, Shared & Valid is Clear.
Value of Tag Control Dirty Bit is Clear.
Value of Tag Control Shared Bit is Clear.
Value of Tag Control Valid Bit is Set.
Value of Parity Bit Covering Tag Store
Address Bits is Clear.
Tag Address<38:20> Is:  x0000000000000061C

Ext Interface Address Reg xFFFFFFF006000050F
Fill Syndrome Reg        x00000000000000C0C
Ext Interface Status Reg  xFFFFFFF005FFFFFFF
Error Occurred During D-ref Fill
LD LOCK                  xFFFFFFF00002006FF

** IOD SUBPACKET -> **                                IOD 0 Register Subpacket

WHOAMI                   x000002FA  Module Revision  0.
                           VCTY ASIC Rev = 1
                           Bcache Size = 4MB
                           CPU = 0

This Bus Bridge Phy Addr  x000000F9E0000000
                           IOD# 0
Dev Type & Rev Register    x0600A332  CAP Chip Revision:      x00000002
                           Host to PCI Revision:      x00000003
                           I/O Backplane Revision:    x00000003
                           PCI-EISA Bus Bridge Present on PCI
                           Device Class: Host Bus to PCI Bridg MC-PCI
Command Register          x46480FF1  Module Self-Test Passed LED On.
                           Delayed PCI Bus Reads Protocol: Enabled
                           Bridge to PCI Transactions:   Enabled
                           Bridge REQUESTS 64 Bit Data Transactions
                           Bridge ACCEPTS 64 Bit Data Transactions
                           PCI Address Parity Check:     Enabled
                           MC Bus CMD/Addr Parity Check:  Enabled
                           MC Bus NXM Check:             Enabled
                           Check ALL Transactions for Errors
                           Use MC_BMSK for 16 Byte Align Blk Mem Wrt
                           Wrt PEND_NUM Threshold: 8.
                           RD_TYPE Memory Prefetch Algorithm: Short
                           RL_TYPE Mem Rd Line Prefetch Type: Medium
                           RM_TYPE Mem Rd Multiple Cmd Type: Long
                           ARB_MODE PCI Arbitration: Round Robin

Mem Host Address Ext Reg  x00000000  HAE Sparse Mem Adr<31:27> x00000000
IO Host Adr Ext Register  x00000000  PCI Upper Adr Bits<31:25> x00000000
Interrupt Ctrl Register   x00000003  Write Device Interrupt Info Struct:Enabled

```

```

Interrupt Request      x00000000  Interrupts asserted  x00000000
Interrupt Mask0 Register  x00C50110
Interrupt Mask1 Register  x00000000
MC Error Info Register 0  xE0000000

MC Error Info Register 1  x000E88FD  MC Bus Trans Addr<31:4>: E0000000
                                   MC bus trans addr <39:32> x000000FD
                                   MC Command is Read0-IO
                                   CPU0 Master at Time of Error
                                   Device ID 2  x00000002

CAP Error Register      x00000000  ④
PCI Bus Trans Error Adr  x00000000
MDPA Status Register    x00000000  MDPA Status Register Data Not Valid
MDPA Error Syndrome Reg x00000000  MDPA Syndrome Register Data Not Valid
MDPB Status Register    x00000000  MDPB Status Register Data Not Valid
MDPB Error Syndrome Reg x00000000  MDPB Syndrome Register Data Not Valid

** IOD SUBPACKET -> **      IOD 1 Register Subpacket

WHOAMI                  x000002FA  Module Revision  0.
                                   VCTY ASIC Rev = 1
                                   Bcache Size = 4MB
                                   CPU = 0

This Bus Bridge Phy Addr x000000FBE0000000
                                   IOD# 1
Dev Type & Rev Register  x06002332  CAP Chip Revision:      x00000002
                                   Host to PCI Revision:      x00000003
                                   I/O Backplane Revision:     x00000003
                                   Internal CAP Chip Arbiter: Enabled
                                   Device Class: Host Bus to PCI Bridg
MC-PCI Command Register  x46480FF1  Module Self-Test Passed LED On.
                                   Delayed PCI Bus Reads Protocol: Enabled
                                   Bridge to PCI Transactions: Enabled
                                   Bridge REQUESTS 64 Bit Data Transactions
                                   Bridge ACCEPTS 64 Bit Data Transactions
                                   PCI Address Parity Check: Enabled
                                   MC Bus CMD/Addr Parity Check: Enabled
                                   MC Bus NXM Check: Enabled
                                   Check ALL Transactions for Errors
                                   Use MC_BMSK for 16 Byte Align Blk Mem Wrt
                                   Wrt PEND_NUM Threshold: 8.
                                   RD_TYPE Memory Prefetch Algorithm: Short
                                   RL_TYPE Mem Rd Line Prefetch Type: Medium
                                   RM_TYPE Mem Rd Multiple Cmd Type: Long
                                   ARB_MODE PCI Arbitration: Round Robin
Mem Host Address Ext Reg x00000000  HAE Sparse Mem Adr<31:27> x00000000
IO Host Adr Ext Register x00000000  PCI Upper Adr Bits<31:25> x00000000
Interrupt Ctrl Register  x00000003  Write Device Interrupt Info Struct:Enabled
Interrupt Request        x00800000  Interrupts asserted  x00000000
                                   Hard Error

Interrupt Mask0 Register x00C50111
Interrupt Mask1 Register x00000000
MC Error Info Register 0  xE0000000

MC Error Info Register 1  x000E88FD  MC Bus Trans Addr<31:4>: E0000000
                                   MC bus trans addr <39:32> x000000FD
                                   MC Command is Read0-IO
                                   CPU0 Master at Time of Error
                                   Device ID 2  x00000002  NOT VALID! ⑥

CAP Error Register      x00000012  Serious error ⑤
                                   PCI error address reg locked

PCI Bus Trans Error Adr  xC157B5C0
MDPA Status Register    x00000000  MDPA Status Register Data Not Valid

```

MDPA Error Syndrome Reg	x00000000	MDPA Syndrome Register Data Not Valid
MDPB Status Register	x00000000	MDPB Status Register Data Not Valid
MDPB Error Syndrome Reg	x00000000	MDPB Syndrome Register Data Not Valid

PALcode Revision	Palcode Rev: 1.21-20
------------------	----------------------

** PCI SUBPACKET -> **	PCI 1 Subpacket
----------------------------------	------------------------

Node Qty	4.
----------	----

CONFIG Address	x000000FBC0000800	Slot or Device Number: 1
Device and Vendor ID	x00011000	NCR 53C810 NCR_810 SCSI Narrow SingleEnded
		Vendor ID: x1000 (NCR)
		Device ID: x00000001
Command Register	x0147	I/O Space Accesses Response: Enabled Memory Space Accesses Response: Enabled PCI Bus Master Capability: Enabled Monitor for Special Cycle Ops: DISABLED Generate Mem Wrt/Invalidate Cmds:DISABLED Parity Error Detection Response: Normal Wait Cycle Address/Data Stepping:DISABLED SERR# Sys Err Driver Capability: Enabled Fast Back-to-Back to Many Target:DISABLED
Status Register	x0200	Device is 33 Mhz Capable. ⑦ No Support for User Defineable Features. Fast Back-to-Back to Different Targets, Is Not Supported in Target Device. Device Select Timing: Medium.

Revision ID	x02	
Device Class Code	x010000	Mass Storage: SCSI Bus Controller
Cache Line S	x00	
Latency T.	xFF	
Header Type	x00	Single Function Device
Bist	x00	
Base Address Register 1	x00101200	
Base Address Register 2	x0412A100	
Base Address Register 3	x00000000	
Base Address Register 4	x00000000	
Base Address Register 5	x00000000	
Base Address Register 6	x00000000	
Expansion Rom Base Address	x00000000	
Interrupt P1	x04	
Interrupt P2	x01	
Min Gnt	x00	
Max Lat	x00	

CONFIG Address	x000000FBC0001000	Slot or Device Number: 2
Device and Vendor ID	x10201077	QLogic ISP_1020
		Vendor ID: x102B (QLogic)
		Device ID: x00001020
Command Register	x0107	I/O Space Accesses Response: Enabled Memory Space Accesses Response: Enabled PCI Bus Master Capability: Enabled Monitor for Special Cycle Ops: DISABLED Generate Mem Wrt/Invalidate Cmds: DISABLED Parity Error Detection Response: *IGNORE* Wait Cycle Address/Data Stepping: DISABLED SERR# Sys Err Driver Capability: Enabled Fast Back-to-Back to Many Target: DISABLED


```

Status Register          x0200 Device is 33 Mhz Capable. ⑦
                          No Support for User Defineable Features.
                          Fast Back-to-Back to Different Targets,
                          Is Not Supported in Target Device.
                          Device Select Timing: Medium.

Revision ID              x05
Device Class Code        x010000 Mass Storage: SCSI Bus Controller
Cache Line S             x10
Latency T.               xF8
Header Type              x00 Single Function Device
Bist                     x00
Base Address Register 1   x00101100
Base Address Register 2   x04129000
Base Address Register 3   x00000000
Base Address Register 4   x00000000
Base Address Register 5   x00000000
Base Address Register 6   x00000000
Expansion Rom Base Address  x04110000
Interrupt P1              x08
Interrupt P2              x01
Min Gnt                   x00
Max Lat                   x00

CONFIG Address           x000000FBC0001800
                          Slot or Device Number: 3
Device and Vendor ID      x00011069 Mylex DAC960 KZPSC RAID Controller
                          Vendor ID: x1069 (Mylex)
                          Device ID: x00000001
Command Register          x0107 I/O Space Accesses Response: Enabled
                          Memory Space Accesses Response: Enabled
                          PCI Bus Master Capability: Enabled
                          Monitor for Special Cycle Ops: DISABLED
                          Generate Mem Wrt/Invalidate Cmds: DISABLED
                          Parity Error Detection Response: *IGNORE*
                          Wait Cycle Address/Data Stepping: DISABLED
                          SEERR# Sys Err Driver Capability: Enabled
                          Fast Back-to-Back to Many Target: DISABLED

Status Register          x8200 Device is 33 Mhz Capable. ⑦
                          No Support for User Defineable Features.
                          Fast Back-to-Back to Different Targets,
                          Is Not Supported in Target Device.
                          Device Select Timing: Medium.
                          DETECTED PARITY ERROR:This Device Detected

Revision ID              x02
Device Class Code        x010400 Mass Storage: RAID Controller
Cache Line S             x10
Latency T.               xFF
Header Type              x00 Single Function Device
Bist                     x00
Base Address Register 1   x00101000
Base Address Register 2   x0412A000
Base Address Register 3   x00000000
Base Address Register 4   x00000000
Base Address Register 5   x00000000
Base Address Register 6   x00000000
Expansion Rom Base Address  x04120000
Interrupt P1              x0C
Interrupt P2              x01
Min Gnt                   x04
Max Lat                   x00

```

CONFIG Address	x000000FBC0002000	Slot or Device Number: 4
Device and Vendor ID	x00081011	DEC_KZPSA Fast-Wide-Differential SCSI
		Vendor ID: x1011 (Digital Equip Corp)
		Device ID: x00000008
Command Register	x0107	I/O Space Accesses Response: Enabled Memory Space Accesses Response: Enabled PCI Bus Master Capability: Enabled Monitor for Special Cycle Ops: DISABLED Generate Mem Wrt/Invalidate Cmds: DISABLED Parity Error Detection Response: *IGNORE* Wait Cycle Address/Data Stepping: DISABLED SERR# Sys Err Driver Capability: Enabled Fast Back-to-Back to Many Target: DISABLED
Status Register	xA2C0	Device is 33 Mhz Capable. Device Supports User Defineable Features. Fast Back-to-Back to Different Targets, Is Supported in Target Device. Device Select Timing: Medium. RECEIVED MASTER-ABORT:Master Sets When Its Transaction Terminated by MasterAbort. DETECTED PARITY ERROR:This Device Detected
Revision ID	x00	
Device Class Code	x010000	Mass Storage: SCSI Bus Controller
Cache Line S	x10	
Latency T.	xFF	
Header Type	x00	Single Function Device
Bist	x80	
Base Address Register 1	x04128000	
Base Address Register 2	x00000000	
Base Address Register 3	x00100000	
Base Address Register 4	x04000000	
Base Address Register 5	x00000000	
Base Address Register 6	x00000000	
Expansion Rom Base Address	x04100000	
Interrupt P1	x10	
Interrupt P2	x01	
Min Gnt	x08	
Max Lat	x7F	

4.3.6 MCHK 630 Correctable CPU Error

The error log in Example 4–6 shows the following:

- ❶ CPU0 logged the error in a system with two CPUs.
- ❷ During a D-ref fill, the External Interface Status Register shows no error but states that the “data source is b-cache.” (When a CPU chip does not find data it needs to perform a task in any of its caches, it requests data from off the chip to fill its D-cache. It performs a D-ref fill.)
- ❸ Both IOD CAP Error Registers logged no error.
- ❹ The FIL Syndrome Register has a valid ECC code for the lower half of the data.

Machine check 630s are detected by CPUs when they either take data off the system bus or when they access their own B-cache. In this case, the data did not come from the system bus, otherwise bit <30> would be set in the External Interface Status Register. CPU0 had a single-bit, ECC correctable error.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-6 MCHK 630 Correctable CPU Error

```

Logging OS                2. DIGITAL UNIX
System Architecture        2. Alpha 4000/1200 Series
Event sequence number      415.
Timestamp of occurrence    15-JUN-1997 14:56:30
Host name                  whip16

System type register       x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)    x00000002

CPU logging event (mperr) x00000000 ❶

Event validity             1. O/S claims event is valid
Event severity             3. High Priority
Entry type                100. CPU Machine Check Errors

CPU Minor class           3. Bcache error (630 entry)

Software Flags             x00000000
Active CPUs               x00000003
Hardware Rev              x00000000
System Serial Number      C1563
Module Serial Number
Module Type               x0000
System Revision           x00000000

Machine Check Reason       x0086 Alpha Chip Detected ECC Err, From B-Cache
EI STAT                   xFFFFFFF085FFFFFF

DATA SOURCE IS BCACHE ❷

D-ref fill
EI ADDRESS                 EV56 Chip Rev 5
                          xFFFFFFF00138D85EF
FIL SYNDROME              x000000000000800 ❸
ISR                       x0000000100200000

WHOAMI                    x00000000 Module Revision 0.
                          MID 0.
                          GID 0.

Sys Environmental Regs     x00000000
Base Addr of Bridge       x00000000
Dev Type & Rev Register    x06008021 CAP Chip Revision x00000001
                          Host to PCI Revision x00000003
                          I/O Backplane Revision x00000003
                          PCI-EISA Bus Bridge Present on PCI
                          Device Class: Host bus to PCI Bridg

MC Error Info Register 0 x00000000 MC Bus Trans Addr<31:4>: 0
MC Error Info Register 1 x00000000 MC bus trans addr <39:32> x00000000
                          MC Command is Illegal
                          Illegal
                          Device ID 2 x00000000

CAP Error Register        x00000000 ❹
MDPA Status Register      x00000000 MDPB Status Register Data Not Valid
MDPA Error Syndrome Reg x00000000 MDPB Syndrome Register Data Not Valid
MDPB Status Register      x00000000 MDPB Status Register Data Not Valid
MDPB Error Syndrome Reg x00000000 MDPB Syndrome Register Data Not Valid
PALcode Revision          PALcode Rev: 1.21-3

```

4.3.7 MCHK 620 Correctable Error

The MCHK 620 error is a correctable error detected by the IOD.

The error log in Example 4-7 shows the following:

- ❶ CPU0 logged the error in a system with two CPUs.
- ❷ The External Interface Status Register is not valid.
- ❸ The MC Error Info Registers 0 and 1 captured the error information.
- ❹ The commander at the time of the error was CPU0.
- ❺ The command at the time of the error was a write-back memory command.

The IOD detected a recoverable error on the system bus. The MC command at the time of the error is a Write Back-Mem Command (x00000016). The system bus commander at the time of the error is CPU0. Since this is a write, the defective FRU is CPU0.

NOTE: The error log example has been edited to decrease its size; registers of interest are in bold type. The “MC” bus is the system bus.

Refer to Table 4-9 for information on decoding commands, and refer to Table 4-10 for information on node IDs.

Example 4-7 MCHK 620 Correctable Error

```

Logging OS                2. DIGITAL UNIX
System Architecture       2. Alpha
Event sequence number     32.
Timestamp of occurrence   28-JUN-1997 19:45:42
Host name                 sect06

System type register      x00000016 AlphaServer 4000/1200 Series
Number of CPUs (mpnum)   x00000002
CPU logging event (mperr) x00000000 ❶

Event validity            1. O/S claims event is valid
Event severity            5. Low Priority
Entry type                100. CPU Machine Check Errors

CPU Minor class           4. 620 System Correctable Error

Software Flags            x0000000000000000
Active CPUs               x00000003
Hardware Rev              x00000000
System Serial Number      C1563
Module Serial Number
Module Type                x0000
System Revision            x00000000

Machine Check Reason       x0204 IOD Detected Soft Error
Ext Interface Status Reg  x0000000000000000
                           Not Valid for 620 System ❷

Correctable Errors
Ext Interface Address Reg x0000000000000000
                           Not Valid for 620 System
                           Correctable Errors
Fill Syndrome Reg         x0000000000000000
                           Not Valid for 620 System
                           Correctable Errors
Interrupt Summary Reg     x0000000000000000
                           Not Valid for 620 System
                           Correctable Errors
WHOAMI                    x00000000 Module Revision 0.
                           MID 0.
                           GID 0.

Sys Environmental Regs     x00000000
Base Addr of Bridge       x000000FBE0000000
Dev Type & Rev Register    x06008021 CAP Chip Revision x00000001
                           Host to PCI Revision x00000003
                           I/O Backplane Revision x00000003
                           PCI-EISA Bus Bridge Present on PCI
                           Device Class: Host bus to PCI Bridg

MC Error Info Register 0  x122D5640 MC Bus Trans Addr<31:4>: 122D5640
MC Error Info Register 1  x800E9600 MC bus trans addr <39:32> x00000000
                           MC Command is WriteBack Mem ❸

CPU0 Master at Time of Error

                           Device ID 2 x00000002 ❹
                           MC error info valid
CAP Error Register        x89000000 Error Detected but Not Logged

```

MC error info latched

Correctable ECC err det by MDPA ③

MDPA Status Register	x00000000	MDPA Status Register Data Not Valid
MDPA Error Syndrome Reg	x00000000	MDPA Syndrome Register Data Not Valid
MDPB Status Register	x00000000	MDPB Status Register Data Not Valid
MDPB Error Syndrome Reg	x00000000	MDPB Syndrome Register Data Not Valid
PALcode Revision		Palcode Rev: 0.0-1

4.4 Troubleshooting IOD-Detected Errors

Step 1

Read the CAP Error Registers on both PCI bridges (F9E0000880 and FBE0000880). If one or both of these registers shows an error, match the register contents with the data pattern and perform the action indicated.

Table 4-3 CAP Error Register Data Pattern

Data Pattern	Most Likely Cause	Action
110x x00x x000 0000 0000 000x xxxx	RDSB - Uncorrectable ECC error detected on upper QW of MC bus (D127:64>)	Go to Step 2
101x x00x x000 0000 0000 000x xxxx	RDSA - Uncorrectable ECC error detected on lower QW of MC bus (D63:0>)	Go to Step 2
111x x00x x000 0000 0000 000x xxxx	RDS detected in both QWs	Go to Step 2
1001 1000 x000 0000 0000 000x xxxx	CRDB - Correctable ECC error detected on upper QW of MC bus (D127:64>)	Go to Step 2
1000 0000 x000 0000 0000 000x xxxx	CRDA - Correctable ECC error detected on lower QW of MC bus (D63:0>)	Go to Step 2
1001 1000 x000 0000 0000 000x xxxx	CRD detected in both QWs.	Go to Step 2
100x x10x x000 0000 0000 000x xxxx	NXM - Nonexistent MC bus address	Go to Step 3
100x x01x x000 0000 0000 000x xxxx	MC_ADR_PERR - MC bus address parity error	Go to Step 4
100x x00x 1000 0000 0000 000x xxxx	PIO_OVFL - PIO buffer overflow	Go to Step 5
0000 0000 0000 0000 0000 0001 1xxx	PTE_INV - Page table entry is invalid	Go to Step 6
0000 0000 0000 0000 0000 0001 x1xx	MAB - Master abort	Go to Step 7
0000 0000 0000 0000 0000 0001 xx1x	SERR - PCI system error	Go to Step 8
0000 0000 0000 0000 0000 0001 xxx1	PERR - PCI parity error	Go to Step 9

4.4.1 System Bus ECC Error

Step 2

Read the MC_ERR1 register and match the contents with the data pattern. Perform the action indicated.

Table 4-4 System Bus ECC Error Data Pattern

MC_ERR1 Data Pattern	Most Likely Cause	Action
for Memory Read		
1000 0000 0000 xxxx xxxx 10xx 0xxx xxxx	Bad nondirty data from memory (bad memory)	Go to Step 10
1000 0000 0000 xxxx xxxx 111x 0xxx xxxx	Bad nondirty data from memory (bad memory)	Go to Step 10
1000 0000 0001 xxxx xxxx 10xx 0xxx xxx	Bad dirty data from a CPU	Replace CPU(s)
1000 0000 0001 xxxx xxxx 111x 0xxx xxxx	Bad dirty data from a CPU	Replace CPU(s)
for Memory or I/O Write		
1000 0000 000x xxx0 10xx 011x xxxx xxxx	Bad data from MID = 2	Replace CPU0
1000 0000 000x xxx0 11xx 011x xxxx xxxx	Bad data from MID = 3	Replace CPU1
1000 0000 000x xxx1 00xx 011x xxxx xxxx	Bad data from MID = 4	Replace Mbrd
1000 0000 000x xxx1 01xx 011x xxxx xxxx	Bad data from MID = 5	Replace Mbrd
for Memory Fill Transactions		
1000 0000 000x xxx1 00xx 110x xxxx xxxx	Bad data from MID = 4	Replace Mbrd
1000 0000 000x xxx1 01xx 110x xxxx xxxx	Bad data from MID = 5	Replace Mbrd
1000 0000 000x xxx1 10xx 110x xxxx xxxx	Bad data from MID = 6	Replace Mbrd
1000 0000 000x xxx1 11xx 110x xxxx xxxx	Bad data from MID = 7	Replace Mbrd

4.4.2 System Bus Nonexistent Address Error

Step 3

Determine which node (if any) should have responded to the command/address identified in MC_ERR1. Perform the action indicated.

Table 4-5 System Bus Nonexistent Address Error Troubleshooting

MC_ERR1 Data Pattern	Most Likely Cause	Action
1000000000xxxxxxx0xxx	Software generated an MC ADDR > TOP_OF_MEM reg	Fix software
1000000000xxxxxxx1xx100x	PCI0 bridge did not respond	Replace Mbrd
1000000000xxxxxxx1xx101x	PCI1 bridge did not respond	Replace Mbrd
1000000000xxxxxxx1xx110x	PCI2 bridge did not respond	Replace Mbrd
1000000000xxxxxxx1xx111x	PCI3 bridge did not respond	Replace Mbrd

4.4.3 System Bus Address Parity Error

Step 4

Determine which node put the bad command/address on the system bus identified in MC_ERR1. Perform the action indicated.

Table 4-6 Address Parity Error Troubleshooting

MC_ERR1 Data Pattern	Most Likely Cause	Action
1000000000x xxx0 10xx xxxx xxxx xxxx	Data sourced by MID = 2	Replace CPU0
1000000000x xxx0 11xx xxxx xxxx xxxx	Data sourced by MID = 3	Replace CPU1
1000000000x xxx1 00xx xxxx xxxx xxxx	Data sourced by MID = 4	Replace Mbrd
1000000000x xxx1 01xx xxxx xxxx xxxx	Data sourced by MID = 5	Replace Mbrd

4.4.4 PIO Buffer Overflow Error (PIO_OVFL)

Step 5

Enter the value of the CAP_CTRL register bits<19:16> (Actual_PEND_NUM) in the following formula. Compare the results as indicated in Table 4-7 to determine the most likely cause of the error. When an IOD is implicated in the analysis of the error, replace the one that captured the error in its CAP Error Register.

$$\text{Expected_PEND_NUM} = 12 - ((2 * (X - 1)) + Y)$$

Where: X = Number of PCIs

Y = Number of CPUs

Table 4-7 Cause of PIO_OVFL Error

Comparison	Most Likely Cause	Action
Actual_PEND_NUM = Expected_PEND_NUM	Broken hardware on IOD	Replace Mbrd
Actual_PEND_NUM < Expected_PEND_NUM	Broken hardware on IOD	Replace Mbrd
Actual_PEND_NUM > Expected_PEND_NUM	PEND_NUM setup incorrect	Fix the software

4.4.5 Page Table Entry Invalid Error

Step 6

This error is almost always a software problem. However, if the software is known to be good and the hardware is suspected, swap the motherboard.

4.4.6 PCI Master Abort

Step 7

Master aborts normally occur when the operating system is sizing the PCI bus. However, if the master abort occurs after the system is booted, read PCI_ERR1 and determine which PCI device should have responded to this PCI address. Replace this device.

4.4.7 PCI System Error

Step 8

For this error to occur a PCI device asserted SERR. Read the error registers in all the PCI devices to determine which device. The PCI device that set SERR should have information logged in its error registers that should indicate a device.

4.4.8 PCI Parity Error

Step 9

Read PCI_ERR1 and determine which PCI device normally uses that PCI address space. Replace that device. Also, read the error registers in all the PCI devices to determine which device was driving the PCI bus when the parity error occurred.

4.4.9 Broken Memory

Step 10

Refer to the following sections.

For a Read Data Substitute Error (uncorrectable ECC error)

When a read data substitute (RDS) error occurs, determine which memory module pair caused the error as follows:

1. Run the memory diagnostic to see if it catches the bad memory. If so, replace the memory module that it reports as bad.
2. At the SRM console prompt, enter the **show mem** command.

```
P00>>> show mem
```

This command displays the base address and size of the memory module pair for each slot.

OR

Read the configuration packet, found in the error log, to retrieve the base address and size of the memory module pair.

3. Compare this address to the failing address from the MC_ERR1 and MC_ERR0 Registers to determine which memory slot is failing.
4. Replace both memory modules (high and low) for that slot. For an RDS error, there is no way to know which memory module (high or low) is bad.

For a Corrected Read Data Error (CRD)

When a CRD error occurs, determine which memory module pair caused the error as follows:

1. At the SRM console prompt, enter the **show mem** command. This command displays the base address and size of the memory module pair for each slot.

```
P00>>> show mem
```

2. Compare this address to the failing address from the MC_ERR1 and MC_ERR0 Registers to determine which memory slot is failing.

3. When you have isolated the failing memory pair, determine which of the two DIMMs is bad. (You cannot do this if the operating system is Windows NT.) Read the CPU FIL SYNDROME Register. If this register is non-zero, use the ECC syndrome bits in Table 4-8 to determine which DIMM had the single-bit error.

Table 4-8 ECC Syndrome Bits Table

CPU Syndrome Values for Low-Order Memory								
01	02	04	08	10	20	40	80	CE
CB	D3	D5	D6	D9	DA	DC	23	25
26	29	2C	31	13	19	4F	4A	52
54	57	58	5B	5D	A2	A4	A8	B0
CPU Syndrome Values for High-Order Memory								
2A	34	0E	0B	15	16	1A	1C	E3
E5	E6	E9	EA	EC	F1	F4	A7	AB
AD	B5	8F	8A	92	94	97	98	9B
9D	62	64	67	68	6B	6D	70	75

4.4.10 Command Codes

Table 4-9 shows the command codes for transactions on the system bus. Note that they are affected by the commander in charge of the bus during the transaction. The command is a six-bit field in the command address (bits<5:0>). Bit-to-text translations give six-bit data (the top two bits may or may not be relevant). Note that address bit<39> defines the command as being either a system space or an I/O command.

Table 4-9 Decoding Commands

5 4	MC_CMD 3 2 1 0	CMD in Hex	MC_ADR <39>	Description	IOD
x x	0 0 0 0	X 0	1	Mem Idle	Y
0 0	0 0 1 0	0 2	1	Write Pend Ack	Y
x x	0 0 1 1	X 3	1	Mem Refresh	
x x	0 1 0 1	X 4	0	Set Dirty	
x 0	0 1 1 0	0/2 6	0	Write Thru - Mem	
x 0	0 1 1 0	0/2 6	1	Write Thru - I/O	
x 1	0 1 1 0	3/1 6	0	Write Back - Mem	
x 1	0 1 1 0	3/1 6	1	Write Intr - I/O	Y
0 0	0 1 1 1	0 7	0	Write Full - Mem	Y
1 0	0 1 1 1	2 7	0	Write Part - Mem	Y
x 0	0 1 1 1	0/2 7	1	Write Mask - I/O	Y
x 0	0 1 1 1	0/2 7	0	Write Merge - Mem	Y
x x	1 0 0 0	X 8	0	Read0 - Mem	Y
x x	1 0 0 0	X 8	1	Read0 - I/O	
x x	1 0 0 1	X 9	0	Read1 - Mem	Y
x x	1 0 0 1	X 9	1	Read1 - I/O	
x x	1 0 1 0	X A	0	Read Mod0 - Mem	Y

Table 4-9 Decoding Commands (continued)

5 4	MC_CMD 3 2 1 0	CMD in Hex	MC_ADR <39>	Description	IOD
x x	1 0 1 0	X A	1	Read Peer0 - I/O	Y
x x	1 0 1 1	X B	0	Read Mod1 - Mem	Y
x x	1 0 1 1	X B	1	Read Peer1 - I/O	Y
1 0	1 1 0 0	2 C	1	FILL0 (due to Read0/Peer0)	Y
1 0	1 1 0 1	2 D	1	FILL1 (due to Read1/Peer1)	Y
x x	1 1 1 0	X E	0	Read0 - Mem	
x x	1 1 1 1	X F	0	Read1 - Mem	

4.4.11 Node IDs

The node ID is a six-bit field in the command address (bits<38:33>). The high-order three bits are always set, and the last three indicate the node. Bit-to-text translations give six-bit data, although only the last three bits define the node.

Table 4-10 Node IDs

Node ID <2:0>	Six Bit (Hex)	Node
0 0 0	38	
0 0 1	39	Memory
0 1 0	3A	CPU0
0 1 1	3B	CPU1
1 0 0	3C	IOD0 on Mbrd
1 0 1	3D	IOD1 on Mbrd
1 1 0	NA	NA
1 1 1	NA	NA

4.5 Double Error Halts and Machine Checks While in PAL Mode

Two error cases require special attention. Neither double error halts or machine checks while the machine is in PAL mode result in error log entries. Nevertheless, information is available that can help determine what error occurred.

4.5.1 PALcode Overview

PALcode, privileged architecture library code, is used to implement a number of functions at the machine level without the use of microcode. This allows operating systems to make common calls to PALcode routines without knowing the hardware specifics of each system the operating system is running on. PALcode routines handle:

- Instructions that require complex sequencing, such as atomic operations
- Instructions that require VAX-style interlocked memory access
- Privileged instructions
- Memory management
- Context swapping
- Interrupt and exception dispatching
- Power-up initialization and booting
- Console functions
- Emulation of instructions with no hardware support

4.5.2 Double Error Halt

A double error halt occurs under the following conditions:

- A machine check occurs.
- PAL completes its tasks and returns control to the operating system.
- A second machine check occurs before the operating system completes its tasks.

The machine returns to the console and displays the following message:

```
halt code = 6
double error halt
PC = 20000004
Your system has halted due to an irrecoverable
error. Record the error halt code and PC and
contact your Digital Services representative. In
addition, type INFO 5 and INFO 8 at the console and
record the results.
```

The **info 5** command (Example 4–9) causes the SRM console to read the PAL-built logout area that contains all the data used by the operating system to create the error entry.

The **info 8** command (Example 4–10) causes the SRM console to read the IOD 0 and IOD 1 registers.

4.5.3 Machine Checks While in PAL

If a machine check occurs while the system is running PALcode, PALcode returns to the SRM console, not to the operating system. The SRM console writes:

```
halt code = 7
machine check while in PAL mode
PC = 20000004
Your system has halted due to an irrecoverable
error. Record the error halt code and PC and contact
your Digital Services representative. In addition, type
INFO 3 and INFO 8 at the console and record the results.
```

The **info 3** command (Example 4–8) causes the SRM console to read the “impure area,” which contains the state of the CPU before it entered PAL.

Example 4–8 INFO 3 Command

```
P00>>> info 3
      cpu00
```

```
per_cpu impure area      00004400
cns$flag                 00000001 : 0000
cns$flag+4               00000000 : 0004
cns$hlt                  00000000 : 0008
cns$hlt+4                 00000000 : 000c
```

cns\$mchkflag	00000228	: 0210
cns\$mchkflag+4	00000000	: 0214
cns\$exc_addr	20000004	: 0318
cns\$exc_addr+4	00000000	: 031c
cns\$pal_base	00000000	: 0320
cns\$pal_base+4	00000000	: 0324
cns\$mm_stat	0000da10	: 0338
cns\$mm_stat+4	00000000	: 033c
cns\$va	00080000	: 0340
cns\$va+4	00000002	: 0344
cns\$iclr	40000000	: 0348
cns\$iclr+4	000000c1	: 034c
cns\$ipl	0000001f	: 0350
cns\$ipl+4	00000000	: 0354
cns\$ps	00000000	: 0358
cns\$ps+4	00000000	: 035c
cns\$itb_asn	00000000	: 0360
cns\$itb_asn+4	00000000	: 0364
cns\$aster	00000000	: 0368
cns\$aster+4	00000000	: 036c
cns\$astrr	00000000	: 0370
cns\$astrr+4	00000000	: 0374
cns\$isir	00400000	: 0378
cns\$isir+4	00000000	: 037c
cns\$ivptbr	00000000	: 0380
cns\$ivptbr+4	00000002	: 0384
cns\$mcsr	00000000	: 0388
cns\$mcsr+4	00000000	: 038c
cns\$dc_mode	00000001	: 0390
cns\$dc_mode+4	00000000	: 0394
cns\$maf_mode	00000080	: 0398
cns\$maf_mode+4	00000000	: 039c
cns\$sirr	00000000	: 03a0
cns\$sirr+4	00000000	: 03a4
cns\$fpcsr	00000000	: 03a8
cns\$fpcsr+4	ff900000	: 03ac
cns\$icperr_stat	00000000	: 03b0
cns\$icperr_stat+4	00000000	: 03b4
cns\$pmctr	00000000	: 03b8
cns\$pmctr+4	00000000	: 03bc
cns\$exc_sum	00000000	: 03c0
cns\$exc_sum+4	00000000	: 03c4
cns\$exc_mask	00000000	: 03c8
cns\$exc_mask+4	00000000	: 03cc
cns\$intid	00000016	: 03d0
cns\$intid+4	00000000	: 03d4
cns\$dcperr_stat	00000000	: 03d8
cns\$dcperr_stat+4	00000000	: 03dc
cns\$sc_stat	00000000	: 03e0
cns\$sc_stat+4	00000000	: 03e4
cns\$sc_addr	000047cf	: 03e8
cns\$sc_addr+4	ffffff00	: 03ec
cns\$sc_ctl	0000f000	: 03f0
cns\$sc_ctl+4	00000000	: 03f4
cns\$bc_tag_addr	ff7fefff	: 03f8
cns\$bc_tag_addr+4	ffffffff	: 03fc
cns\$ei_stat	04ffffff	: 0400
cns\$ei_stat+4	ffffff00	: 0404
cns\$fill_syn	000000a7	: 0410
cns\$fill_syn+4	00000000	: 0414
cns\$ld_lock	0004eaeef	: 0418
cns\$ld_lock+4	ffffff00	: 041c

Example 4-9 INFO 5 Command

```
P00>>> info 5
      cpu00
```

```
per_cpu_logout_area      00004838
mchk$crd_flag            00000320 : 0000
mchk$crd_flag+4          00000000 : 0004
mchk$crd_offsets         00000118 : 0008
mchk$crd_offsets+4       00001328 : 000c
mchk$crd_mchk_code       00980000 : 0010
mchk$crd_mchk_code+4     00000000 : 0014
mchk$crd_ei_stat         eba00003 : 0018
mchk$crd_ei_stat+4       4143040a : 001c
mchk$crd_ei_addr         d1200067 : 0020
mchk$crd_ei_addr+4       47f90416 : 0024
mchk$crd_fill_syn        eba00003 : 0028
mchk$crd_fill_syn+4      d1200068 : 002c
mchk$crd_isr             7ec38000 : 0030
mchk$crd_isr+4           63ff4000 : 0034
mchk$flag                00000320 : 0000
mchk$flag+4              00000000 : 0004
mchk$isr                 00000000 : 0138
mchk$isr+4               00000000 : 013c
mchk$icshr               60000000 : 0140
mchk$icshr+4             000000c1 : 0144
mchk$ic_perr_stat        00000000 : 0148
mchk$ic_perr_stat+4      00000000 : 014c
mchk$dc_perr_stat        00000000 : 0150
mchk$dc_perr_stat+4      00000000 : 0154
mchk$va                  ff8000a0 : 0158
mchk$va+4                ffffffff : 015c
mchk$nm_stat             000149d0 : 0160
mchk$nm_stat+4           00000000 : 0164
mchk$sc_addr             0001904f : 0168
mchk$sc_addr+4           ffffffff00 : 016c
mchk$sc_stat             00000000 : 0170
mchk$sc_stat+4           00000000 : 0174
mchk$bc_tag_addr         ff7fefff : 0178
mchk$bc_tag_addr+4       ffffffff : 017c
mchk$ei_addr             066bc3ef : 0180
mchk$ei_addr+4           ffffffff00 : 0184
mchk$fill_syn            000000a7 : 0188
mchk$fill_syn+4          00000000 : 018c
mchk$ei_stat             04fffffff : 0190
mchk$ei_stat+4           ffffffff00 : 0194
mchk$ld_lock             00005b6f : 0198
mchk$ld_lock+4           ffffffff00 : 019c
```

```
IOD: 0 base address: f9e000000
```

WHOAMI:	0000003a	PCI_REV:	06008221		
CAP_CTL:	02490fb1	HAE_MEM:	00000000	HAE_IO:	00000000
INT_CTL:	00000003	INT_REQ:	00800000	INT_MASK0:	00010000
INT_MASK1:	00000000	MC_ERR0:	e0000000	MC_ERR1:	800e88fd
CAP_ERR:	84000000	PCI_ERR:	00000000	MDPA_STAT:	00000000
MDPA_SYN:	00000000	MDPB_STAT:	00000000	MDPB_SYN:	00000000

```
IOD: 1 base address: fbe000000
```

WHOAMI:	0000003a	PCI_REV:	06000221
----------------	-----------------	-----------------	----------

CAP_CTL:	02490fb1	HAE_MEM:	00000000	HAE_IO:	00000000
INT_CTL:	00000003	INT_REQ:	00800000	INT_MASK0:	00010000
INT_MASK1:	00000000	MC_ERR0:	e0000000	MC_ERR1:	800e88fd
CAP_ERR:	84000000	PCI_ERR:	00000000	MDPA_STAT:	00000000
MDPA_SYN:	00000000	MDPB_STAT:	00000000	MDPB_SYN:	00000000

Example 4-10 INFO 8 Command

P00>>> info 8
IOD 0

WHOAMI:	0000003a	PCI_REV:	06008221		
CAP_CTL:	02490fb1	HAE_MEM:	00000000	HAE_IO:	00000000
INT_CTL:	00000003	INT_REQ:	00000000	INT_MASK0:	00210000
INT_MASK1:	00000000	MC_ERR0:	e0000000	MC_ERR1:	000e88fd
CAP_ERR:	00000000	PCI_ERR:	00000000	MDPA_STAT:	00000000
MDPA_SYN:	00000000	MDPB_STAT:	00000000	MDPB_SYN:	00000000
INT_TARG:	0000003a	INT_ADR:	00006000	INT_ADR_EXT:	00000000
PERF_MON:	00406ebf	PERF_CONT:	00000000	CAP_DIAG:	00000000
DIAG_CHKA:	10000000	DIAG_CHKB:	10000000	SCRATCH:	21011131
W0_BASE:	00100001	W0_MASK:	00000000	T0_BASE:	00001000
W1_BASE:	00800001	W1_MASK:	00700000	T1_BASE:	00008000
W2_BASE:	80000000	W2_MASK:	3ff00000	T2_BASE:	00000000
W3_BASE:	00000000	W3_MASK:	1ff00000	T3_BASE:	0000b800
W_DAC:	00000000	SG_TBIA:	00000000	HBASE:	00000000

IOD 1

WHOAMI:	0000003a	PCI_REV:	06000221		
CAP_CTL:	02490fb1	HAE_MEM:	00000000	HAE_IO:	00000000
INT_CTL:	00000003	INT_REQ:	00000000	INT_MASK0:	00000000
INT_MASK1:	00000000	MC_ERR0:	e0000000	MC_ERR1:	000e88fd
CAP_ERR:	00000000	PCI_ERR:	00000000	MDPA_STAT:	00000000
MDPA_SYN:	00000000	MDPB_STAT:	00000000	MDPB_SYN:	00000000
INT_TARG:	0000003a	INT_ADR:	00006000	INT_ADR_EXT:	00000000
PERF_MON:	004e31a6	PERF_CONT:	00000000	CAP_DIAG:	00000000
DIAG_CHKA:	10000000	DIAG_CHKB:	10000000	SCRATCH:	00000000
W0_BASE:	00100001	W0_MASK:	00000000	T0_BASE:	00001000
W1_BASE:	00800001	W1_MASK:	00700000	T1_BASE:	00008000
W2_BASE:	80000001	W2_MASK:	3ff00000	T2_BASE:	00000000
W3_BASE:	00000000	W3_MASK:	1ff00000	T3_BASE:	0000a000
W_DAC:	00000000	SG_TBIA:	00000000	HBASE:	00000000

Chapter 5

Error Registers

This chapter describes the registers used to hold error information. These registers include:

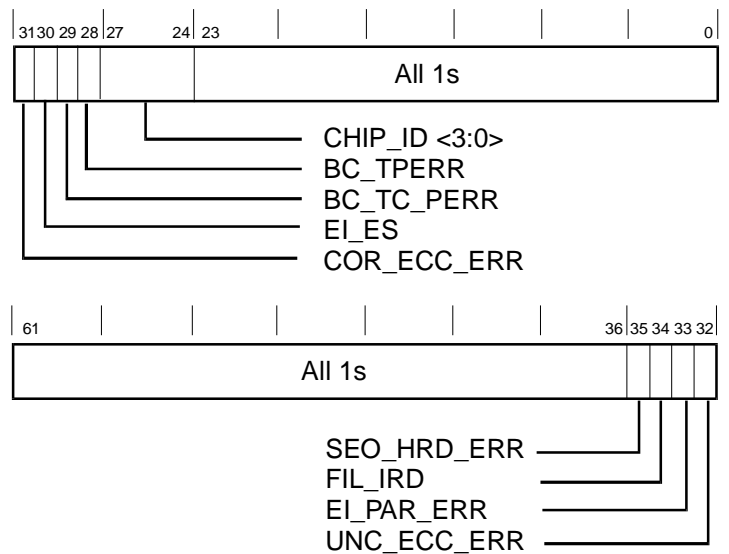
- External Interface Status Register
- External Interface Address Register
- MC Error Information Register 0
- MC Error Information Register 1
- CAP Error Register
- PCI Error Status Register 1

5.1 External Interface Status Register - EI_STAT

The EI_STAT register is a read-only register that is unlocked and cleared by any PALcode read. A read of this register also unlocks the EI_ADDR, BC_TAG_ADDR, and FILL_SYN registers subject to some restrictions. The EI_STAT register is not unlocked or cleared by reset.

Address FF FFF0 0168

Type R



PKW0453-96

Fill data from B-cache or main memory could have correctable or uncorrectable errors in ECC mode. System address/command parity errors are always treated as uncorrectable hard errors, irrespective of the mode. The sequence for reading, unlocking, and clearing EI_STAT, EI_ADDR, BC_TAG_ADDR, and FILL_SYN is as follows:

1. Read the EI_ADDR, BC_TAG_ADDR, and FIL_SYN registers in any order. Does not unlock or clear any register.
2. Read the EI_STAT register. This operation unlocks the EI_ADDR, BC_TAG_ADDR, and FILL_SYN registers. It also unlocks the EI_STAT register subject to conditions given in Table 5-2, which defines the loading and locking rules for external interface registers.

NOTE: If the first error is correctable, the registers are loaded but not locked. On the second correctable error, the registers are neither loaded nor locked.

Registers are locked on the first uncorrectable error except the second hard error bit. This bit is set only for an uncorrectable error that follows an uncorrectable error. A correctable error that follows an uncorrectable error is not logged as a second error. B-cache tag parity errors are uncorrectable in this context.

Table 5-1 External Interface Status Register

Name	Bits	Type	Description
COR_ECC_ERR	<31>	R	Correctable ECC Error. Indicates that fill data received from outside the CPU contained a correctable ECC error.
EI_ES	<30>	R	<p>External Interface Error Source. When set, indicates that the error source is fill data from main memory or a system address/command parity error. When clear, the error source is fill data from the B-cache.</p> <p>This bit is only meaningful when <COR_ECC_ERR>, <UNC_ECC_ERR>, or <EI_PAR_ERR> is set in this register.</p> <p>This bit is not defined for a B-cache tag error (BC_TPERR) or a B-cache tag control parity error (BC_TC_ERR).</p>
BC_TC_PERR	<29>	R	<p>B-Cache Tag Control Parity Error. Indicates that a B-cache read transaction encountered bad parity in the tag control RAM.</p>
BC_TPERR	<28>	R	<p>B-Cache Tag Address Parity Error. Indicates that a B-cache read transaction encountered bad parity in the tag address RAM.</p>
CHIP_ID	<27:24>	R	<p>Chip Identification. Read as “5.” Future update revisions to the chip will return new unique values.</p>
	<23:0>		All ones.

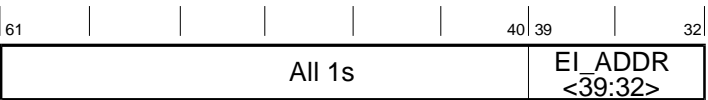
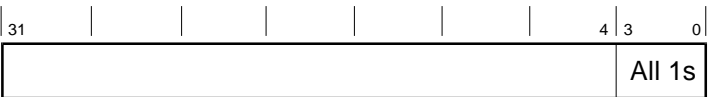
Table 5-1 External Interface Status Register (continued)

Name	Bits	Type	Description
	<63:36>		All ones.
SEO_HRD_ERR	<35>	R	Second External Interface Hard Error. Indicates that a fill from B-cache or main memory, or a system address/command received by the CPU has a hard error while one of the hard error bits in the EI_STST register is already set.
FIL_IRD	<34>	R	Fill I-Ref D-Ref. When set, indicates that the error occurred during an I-ref fill. When clear, indicates that the error occurred during a D-ref fill. This bit has meaning only when one of the ECC or parity error bits is set. This bit is not defined for a B-cache tag parity error (BC_TPERR) or a B-cache tag control parity error (BC_TC_ERR).
EI_PAR_ERR	<33>	R	External Interface Command/Address Parity Error. Indicates that an address and command received by the CPU has a parity error.
UNC_ECC_ERR	<32>	R	Uncorrectable ECC Error. Indicates that fill data received from outside the CPU contained an uncorrectable ECC error. In parity mode, this bit indicates a data parity error.

5.2 External Interface Address Register - EI_ADDR

The EI_ADDR register contains the physical address associated with errors reported by the EI_STAT register. It is unlocked by a read of the EI_STAT Register. This register is meaningful only when one of the error bits is set.

Address FF FFF0 0148
Access R



PKW0454-96

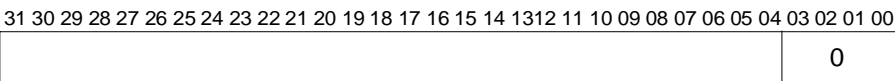
Table 5-2 Loading and Locking Rules for External Interface Registers

Correct-able Error	Uncorrect-able Error	Second Hard Error	Load Register	Lock Register	Action When EI_STAT Is Read
0	0	Not possible	No	No	Clears and unlocks all registers
1	0	Not possible	Yes	No	Clears and unlocks all registers
0	1	0	Yes	Yes	Clears and unlocks all registers
1 ¹	1	0	Yes	Yes	Clear bit (c) does not unlock. Transition to “0,1,0” state.
0	1	1	No	Already locked	Clears and unlocks all registers
1 ¹	1	1	No	Already locked	Clear bit (c) does not unlock. Transition to “0,1,1” state.

¹These are special cases. It is possible that when EI_ADDR is read, only the correctable error bit is set and the registers are not locked. By the time EI_STAT is read, an uncorrectable error is detected and the registers are loaded again and locked. The value of EI_ADDR read earlier is no longer valid. Therefore, for the “1,1,x” case, when EI_STAT is read correctable, the error bit is cleared and the registers are not unlocked or cleared. Software must reexecute the IPR read sequence. On the second read operation, error bits are in “0,1,x” state, all the related IPRs are unlocked, and EI_STAT is cleared.

5.3 MC Error Information Register 0
(MC_ERR0 - Offset = 800)

The low-order MC bus (system bus) address bits are latched into this register when the system bus to PCI bus bridge detects an error event. If the event is a hard error, the register bits are locked. A write to clear symptom bits in the CAP Error Register unlocks this register. When the valid bit (MC_ERR_VALID) in the CAP Error Register is clear, the contents are undefined.



Failing Address ADDR<31:4>

PKW0551-97

Table 5-3 MC Error Information Register 0

Name	Bits	Type	Initial State	Description
ADDR<31:4>	<31:4>	RO	0	Contains the address of the transaction on the system bus when an error is detected.
Reserved	<3:0>	RO	0	

5.4 MC Error Information Register 1 (MC_ERR1 - Offset = 840)

The high-order MC bus (system bus) address bits and error symptoms are latched into this register when the system bus to PCI bus bridge detects an error. If the event is a hard error, the register bits are locked. A write to clear symptom bits in the CAP Error Register unlocks this register. When the valid bit (MC_ERR_VALID) in the CAP Error Register is clear, the contents are undefined.

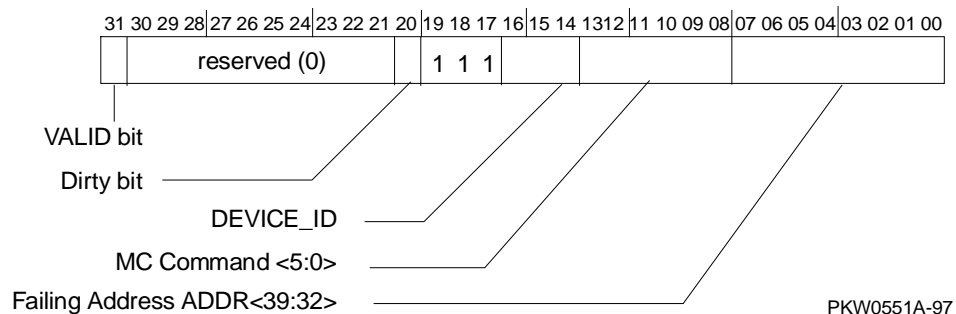


Table 5-4 MC Error Information Register 1

Name	Bits	Type	Initial State	Description
VALID	<31>	RO	0	Logical OR of bits <30:23> in the CAP_ERR Register. Set if MC_ERR0 and MC_ERR1 contain a valid address.
Reserved	<30:21>	RO	0	
Dirty	<20>	RO	0	Set if the system bus error was associated with a Read/Dirty transaction. When set, the device ID field <19:14> does not indicate the source of the data.
Reserved	<19:17>		1	All ones.
DEVICE_ID	<16:14>	RO	0	Slot number of bus master at the time of the error.
MC_CMD<5:0>	<13:8>	RO	0	Active command at the time the error was detected.
ADDR<39:32>	<7:0>	RO	0	Address bits <39:32> of the transaction on the system bus when an error is detected.

5.5 CAP Error Register (CAP_ERR - Offset = 880)

CAP_ERR is used to log information pertaining to an error detected by the CAP or MDP ASIC. If the error is a hard error, the register is locked. All bits, except the **LOST_MC_ERR** bit, are locked on hard errors. **CAP_ERR** remains locked until the CAP error is written to clear each individual error bit.

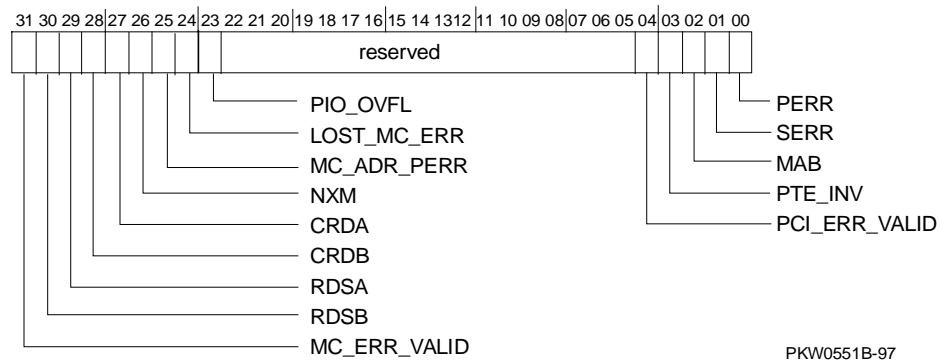


Table 5-5 CAP Error Register

Name	Bits	Type	Initial State	Description
MC_ERR_VALID	<31>	RO	0	Logical OR of bits <30:23> in this register. When set MC_ERR0 and MC_ERR1 are latched.
RDSB	<30>	RW1C	0	Uncorrectable ECC error detected by MDPB. Clear state in MDPB before clearing this bit.
RDSA	<29>	RW1C	0	Uncorrectable ECC error detected by MDPB. Clear state in MDPB before clearing this bit.
CRDB	<28>	RW1C	0	Correctable ECC error detected by MDPB. Clear state in MDPB_STAT before clearing this bit.
CRDA	<27>	RW1C	0	Correctable ECC error detected by MDPB. Clear state in MDPB_STAT before clearing this bit.
NXM	<26>	RW1C	0	System bus master transaction status NXM (Read with Address bit <39> set but transaction not pended or transaction target above the top of memory register.) CPU will also get a fill error on reads.
MC_ADR_PERR	<25>	RW1C	0	Set when a system bus command/address parity error is detected.

Table 5-5 CAP Error Register (continued)

Name	Bits	Type	Initial State	Description
LOST_MC_ERR	<24>	RW1C	0	Set when an error is detected but not logged because the associated symptom fields and registers are locked with the state of an earlier error.
PIO_OVFL	<23>	RW1C	0	Set when a transaction that targets this system bus to PCI bus bridge is not serviced because the buffers are full. This is a symptom of setting the PEND_NUM field in CAP_CNTL to an incorrect value.
Reserved	<22:5>	RO	0	
PCI_ERR_VALID	<4>	RO	0	Logical OR of bits <3:0> of this register. When set, the PCI error address register is locked.
PTE_INV	<3>	RW1C	0	Invalid page table entry on scatter/gather access.
MAB	<2>	RW1C	0	PCI master state machine detected PCI Target Abort (likely cause: NXM) (except Special Cycle). On reads fill error is also returned.
SERR	<1>	RW1C	0	PCI target state machine observed SERR#. CAP asserts SERR when it is master and detects target abort.
PERR	<0>	RW1C	0	PCI master state machine observed PERR#.

5.6 PCI Error Status Register 1 (PCI_ERR1 - Offset = 1040)

PCI_ERR1 is used by the system bus to PCI bus bridge to log bus address <31:0> pertaining to an error condition logged in CAP_ERR. This register always captures PCI address <31:0>, even for a PCI DAC cycle. When the PCI_ERR_VALID bit in CAP_ERR is clear, the contents are undefined.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

Failing Address ADDR<31:0>

PKW0551C-97

Table 5-6 PCI Error Status Register 1

Name	Bits	Type	Initial State	Description
ADDR<31:0>	<31:0>	RO	0	Contains address bits <31:0> of the transaction on the PCI bus when an error is detected.

Chapter 6

Removal and Replacement

This chapter describes removal and replacement procedures for field-replaceable units (FRUs).

6.1 System Safety

Observe the safety guidelines in this section to prevent personal injury.

CAUTION: Wear an antistatic wrist strap whenever you work on a system.

WARNING: When the system interlocks are disabled and the system is still powered on, voltages are low in the system, but current is high. Observe the following guidelines to prevent personal injury.

- 1. Remove any jewelry that may conduct electricity before working on the system.*
- 2. If you need to access the system card cage, power down the system and wait 2 minutes to allow components in that area to cool.*

6.2 FRU List

Figure 6-1 shows the locations of FRUs, and Table 6-1 lists the part numbers of all field-replaceable units.

Figure 6-1 System FRU Locations

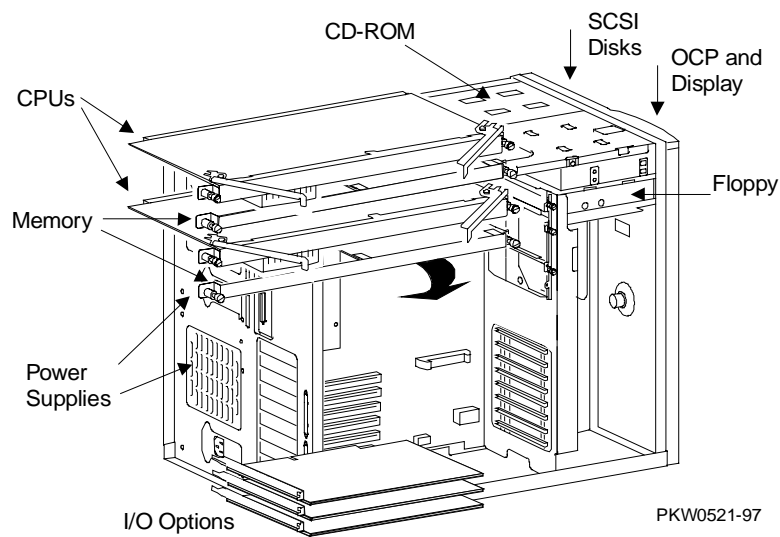


Table 6-1 Field-Replaceable Unit Part Numbers

CPU Modules	
B3007-AA	400 MHz CPU 4 Mbyte cache
B3007-CA	533 MHz CPU, 4 Mbyte cache
Memory Modules	
54-25084-DA 20-47405-D3	32 Mbyte DIMM (synchronous)
54-25092-DA 20-45619-D3	128 Mbyte DIMM (synchronous)
54-25149-01	Memory riser card
System Backplane, Display, and support hardware	
54-25147-01	System motherboard
RX23L-AB	Floppy CD-ROM
54-23302-02	OCP assembly
70-31349-01	Speaker assembly
Fans	
70-31351-01	Cooling fan 120x120
70-31350-01	Cooling fan 92x92
12-24701-34	CPU fan
Power System Components	
30-43120-02	Power supply
SCSI Hardware	
54-23365-01	SCSI backplane Ultra SCSI bus extender

Table 6-1 Field-Replaceable Unit Part Numbers (continued)

Power Cords			
BN26J-1K		North America, Japan 12V, 75-inches long	
BN19H-2E		Australia, New Zealand, 2.5m long	
BN19C-2E		Central Europe, 2.5m long	
BN19A-2E		UK, Ireland, 2.5m long	
BN19E-2E		Switzerland 2.5m long	
BN19K-2E		Denmark, 2.5m long	
BN19Z-2E		Italy, 2.5m long	
BN19S-2E		Egypt, India, South Africa, 2.5m long	
BN18L-2E		Israel, 2.5m long	
Ultra SCSI Cables and Jumpers		From	To
17-04143-01	68 pin con cable	SCSI controller	Ultra SCSI bus extender
17-04022-03	68 pin con cable	Ultra SCSI bus extender	SCSI backpln signal con
17-04021-01	68 pin con jumpr	SCSI backpln	SCSI backpln
17-04019-02	68 pin con cable	External prt on SCSI backpln	Terminator
12-41768-03	68 pin terminator		End or 17-04019-02

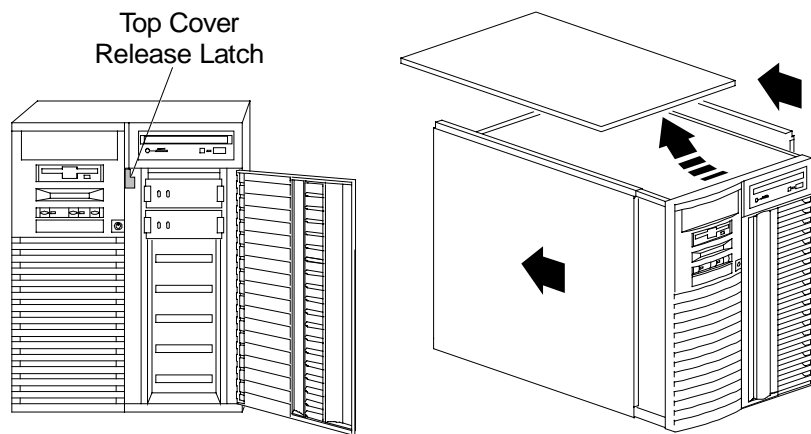
Table 6-1 Field-Replaceable Unit Part Numbers (continued)

System Cables and Jumpers		From	To
17-01495-01	Current share cable	Current share conn on PS0	Current share conn on PS1
17-03970-02	Floppy signal cable (34 pin)	Floppy conn on mbrd	Floppy
17-03971-01	OCP signal	OCP conn on mbrd	OCP signal
	Twisted pair (yellow and green)	J2 RCM conn on mbrd	Power conn on OCP
	Twisted pair (red and black)	OCP	Interlock switch pigtail
70-31348-01	Interlock switch and pigtail cable	Interlock switch assy	Twisted pair (red and black) OCP DC enable pwr cable from OCP conn
17-04685-01	SCSI CD-ROM sig cable	CD-ROM conn on mbrd	CD-ROM sig conn
70-37346-01	Power harness	Power supply(s)	3 conns. On sys mbrd CD-ROM drv pwr Floppy pwr Optional drive above Flop <u>Single ultra SCSI config</u> StorageWorks backpln and pwr cable to Ultra SCSI bus extender <u>Dual Ultra SCSI config</u> two pwr cables to two SCSI bus extenders
17-04700-01	Power cable to Ultra SCSI bus extndr(s) Y cable(s)	Ultra SCSI bus extndr(s) pwr and StrWrks backpln	Power harness

6.3 System Exposure

The system has three sheet metal covers, one on top and one on each side. The covers are removed to expose the system card cage and the power/SCSI sections.

Figure 6-2 Exposing the System



IP00205

Exposing the System

CAUTION: Be sure the system On/Off button is in the “off” position before removing system covers.

1. Shutdown the operating system.
2. Press the On/Off button to turn the system off.
3. Unlock and open the door that exposes the storage shelf.
4. Pull down the top cover latch shown in Figure 6-2 until it latches in the down position.
5. Grasp the finger groove at the rear of the top cover and pull it straight back about 2 inches and then lift it off the cabinet.
6. Pull a side panel back a few inches, tilt the top away from the machine, and lift it off. (Repeat for the other side)

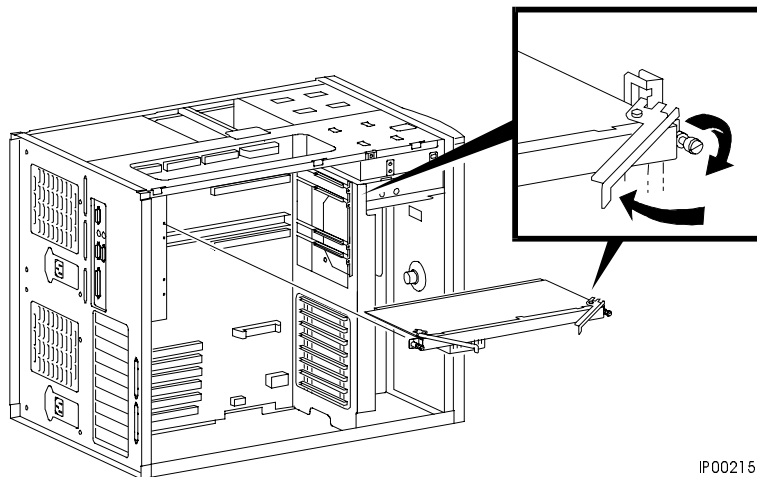
Dressing the System

Reverse the steps in the exposure process.

6.4 CPU Removal and Replacement

CAUTION: Several different CPU modules work in these systems. Unless you are upgrading the system be sure you are replacing the CPU you are removing with the same variant of CPU.

Figure 6-3 Removing CPU Module



IP00215

WARNING: CPU modules and memory modules have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Remove the memory riser card next to the CPU you are removing (see Section 6.6).
4. Loosen the two captive screws holding the module to the card cage.
5. The CPU is held in place with levers at both ends; simultaneously pull the levers away from the module handle and pull the CPU from the cage.

Replacement

Reverse the steps in the Removal procedure.

Verification — DIGITAL UNIX and OpenVMS Systems

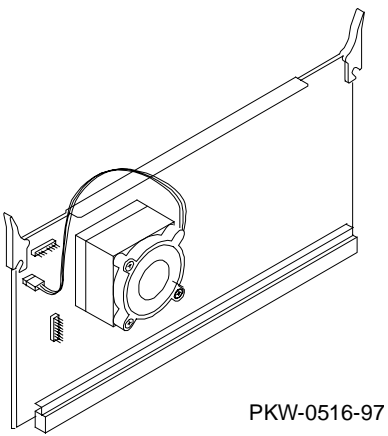
1. Bring the system up to the SRM console by pressing the Halt button, if necessary.
2. Issue the **show cpu** command to display the status of the new module.

Verification — Windows NT Systems

1. Start **AlphaBIOS Setup**, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **MC Bus Configuration** to display the status of the new module.

6.5 CPU Fan Removal and Replacement

Figure 6-4 Removing CPU Fan



Removal

1. Follow the CPU Removal and Replacement procedure.
2. Unplug the fan from the module.
3. Remove the four Phillips head screws holding the fan to the Alpha chip's heatsink.

Replacement

Reverse the above procedure.

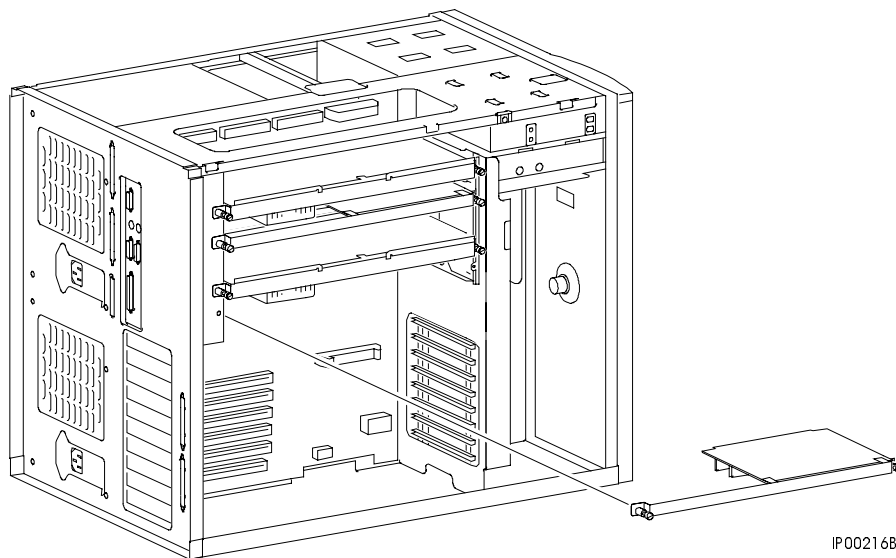
Verification

If the system powers up, the CPU fan is working.

6.6 Memory Riser Card Removal and Replacement

CAUTION: Several different memory DIMMs work in these systems. Be sure you are replacing the broken DIMM with the same variant.

Figure 6-5 Removing Memory Riser Card



WARNING: CPU modules and memory riser cards have parts that operate at high temperatures. Wait 2 minutes after power is removed before touching any module.

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. There are two riser cards, one High and one Low. After you have determined which should be removed, loosen the two captive screws that secure the riser card to the card cage.
4. Lift the riser card from the card cage.

Replacement

Reverse the steps in the Removal procedure.

NOTE: Memory DIMMs are installed in pairs and it is important that the pairs are the same size. When you replace a bad DIMM, be sure to replace it with the same size DIMM as the one you removed.

Verification — DIGITAL UNIX and OpenVMS Systems

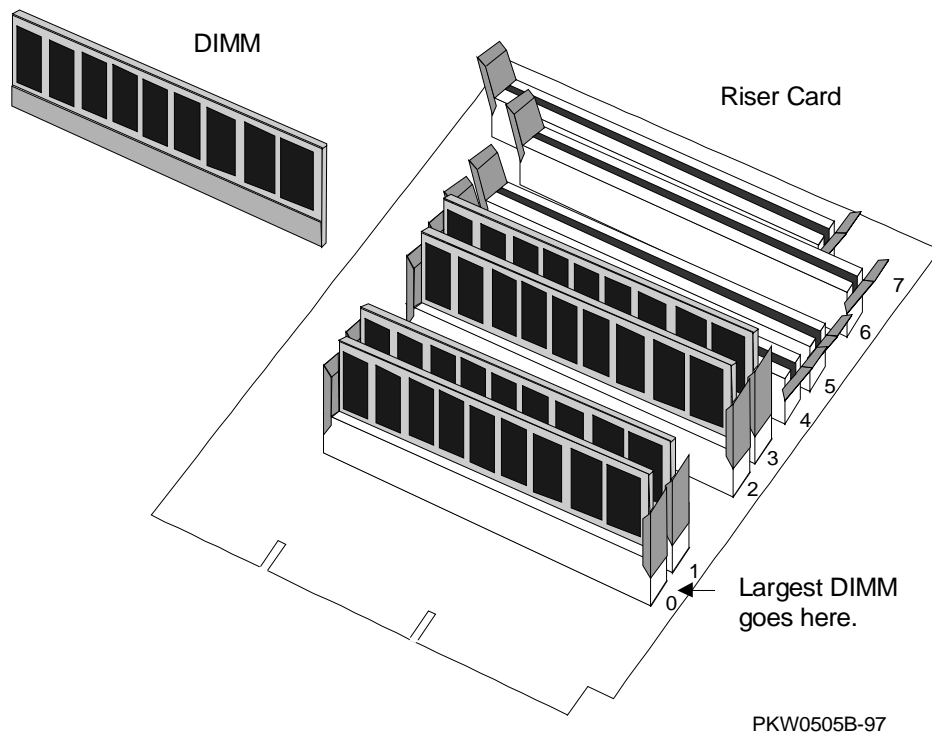
1. Bring the system up to the SRM console by pressing the Halt button, if necessary.
2. Issue the **show memory** command to display the status of the new memory.
3. Verify the functioning of the new memory by issuing the command **test memn**, where *n* is 0, 1, 2, 3, or *.

Verification — Windows NT Systems

1. Start **AlphaBIOS Setup**, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **Memory Configuration** to display the status of the new memory.
3. Switch to the SRM console (press the Halt button in so that the LED on the button lights and reset the system). Verify the functioning of the new memory by issuing the command **test memn**, where *n* is 0, 1, 2, 3, or *.

6.7 DIMM Removal and Replacement

Figure 6-6 Removing a DIMM from a Memory Riser Card



Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Remove the memory riser card that has the broken memory DIMM (see Section 6.6).
4. There are prying/retaining levers on the connectors in each slot on the riser card. Press both levers in an arc away from the DIMM and gently pull the DIMM from the connector.

Replacement

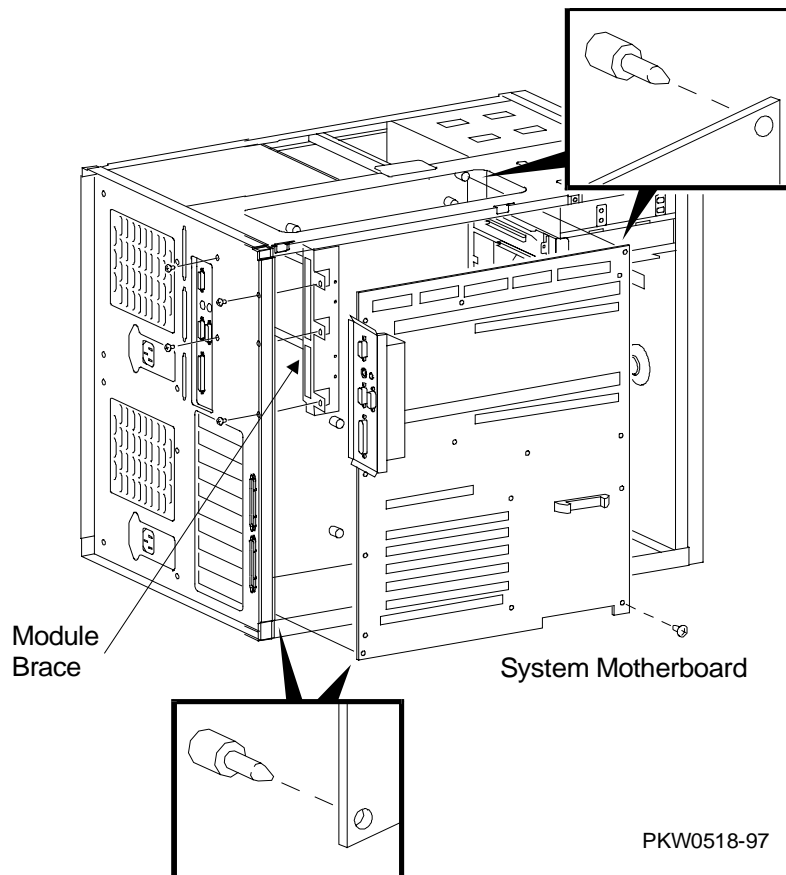
Reverse the steps in the Removal procedure.

Verification

Follow the verification procedure recommended for the memory riser card, Section 6.6.

6.8 System Motherboard Removal and Replacement

Figure 6-7 Removing System Motherboard



Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Remove both memory riser cards.
4. Remove all CPUs.
5. Remove all PCI and EISA options.
6. From the back of the cabinet, using a Phillips head screwdriver, unscrew the four screws holding the CPU and memory riser card brace from the system frame. Remove the brace.
7. Unplug all cables connected to the motherboard and clear access to all screws holding the motherboard in place.
8. Using a Phillips head screwdriver unscrew the eleven screws holding the motherboard in place and remove it from the system. Note the two guide studs, one in the upper right corner and the other in the lower left corner, that protrude through holes in the motherboard.

Replacement

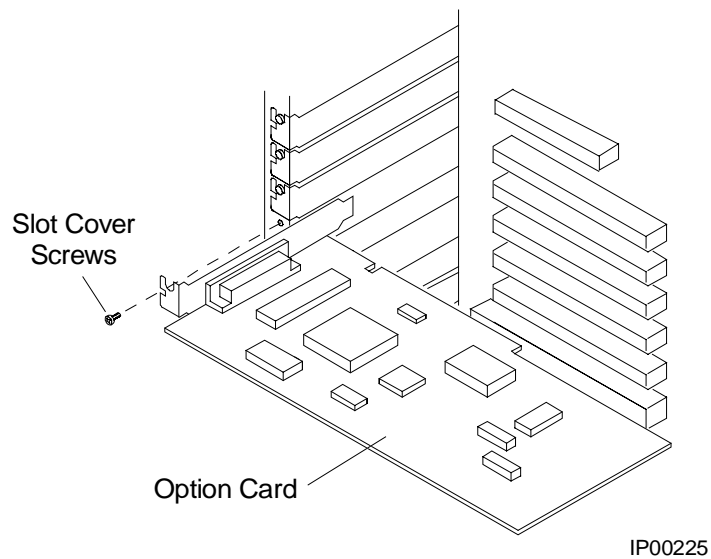
Reverse the steps in the Removal procedure.

Verification

Power up the system (press the Halt button if necessary to bring up the SRM console) and issue the **show device** command at the console prompt to verify that the system sees all system options and peripherals.

6.9 PCI/EISA Option Removal and Replacement

Figure 6-8 Removing PCI/EISA Option



WARNING: To prevent fire, use only modules with current limited outputs. See National Electrical Code NFPA 70 or Safety of Information Technology Equipment, Including Electrical Business Equipment EN 60 950.

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. To remove the faulty option: Disconnect cables connected to the option. Remove cables to other options that obstruct the option you are removing. Unscrew the small Phillips head screw securing the option to the card cage. Slide it from the system.

Replacement

Reverse the steps in the Removal procedure.

Verification — DIGITAL UNIX and OpenVMS Systems

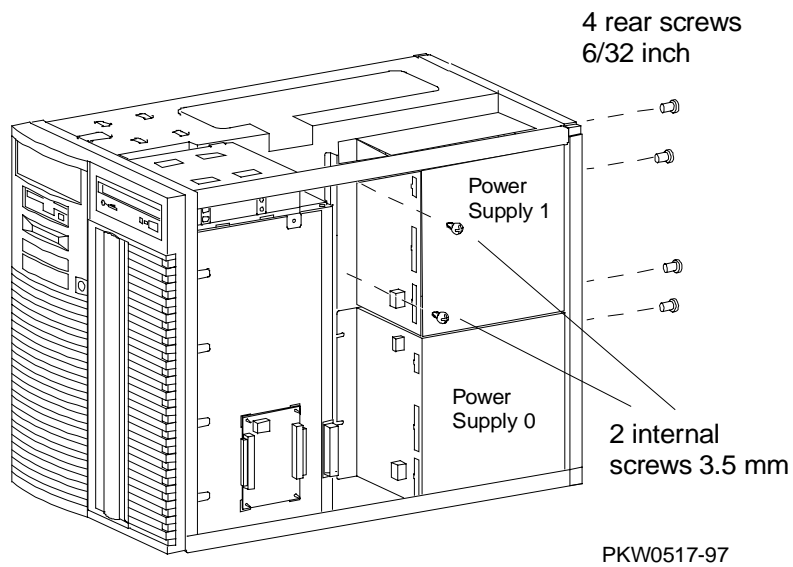
1. Power up the system (press the Halt button if necessary to bring up the SRM console) and run the ECU to restore EISA configuration data.
2. Issue the **show config** command or **show device** command at the console prompt to verify that the system sees the option you replaced.
3. Run any diagnostic appropriate for the option you replaced.

Verification — Windows NT Systems

1. Start **AlphaBIOS Setup**, select **Display System Configuration**, and press Enter.
2. Using the arrow keys, select **PCI Configuration** or **EISA Configuration** to determine that the new option is listed.

6.10 Power Supply Removal and Replacement

Figure 6-9 Removing Power Supply



Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Unplug the power supply you are replacing.
4. Remove the four screws at the back of the system cabinet and the two screws at the back of the power supply that hold the power supply in place.
5. If you are removing power supply 0, slide the supply out the side of the cabinet.
If you are removing power supply 1, lift the supply out the top of the cabinet.

Replacement

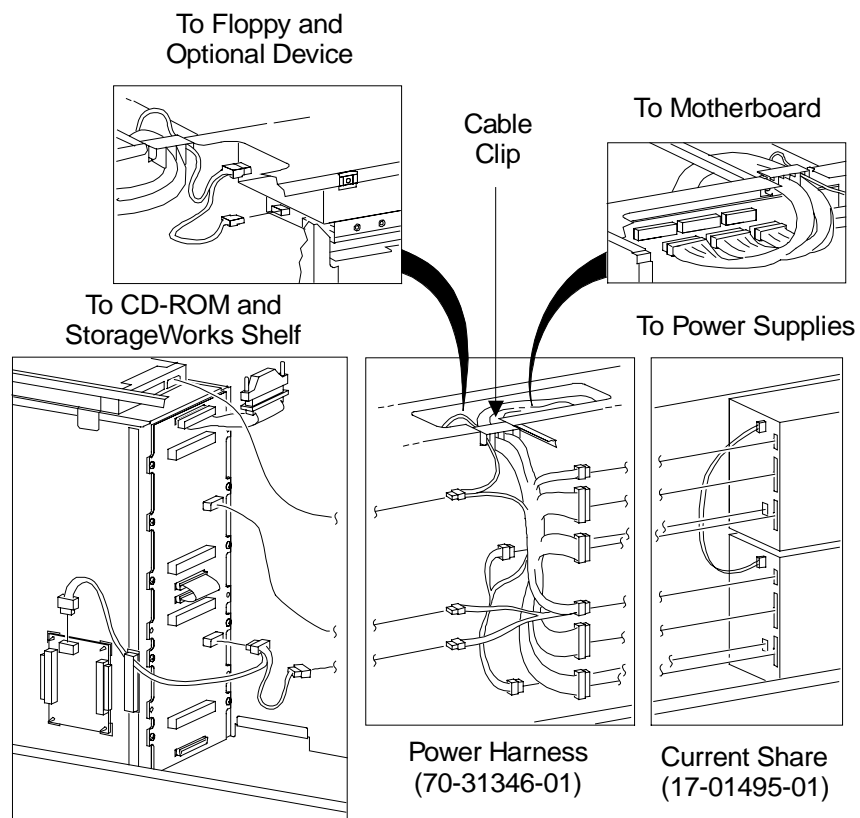
Reverse the steps in the Removal procedure.

Verification

Power up the system.

6.11 Power Harness Removal and Replacement

Figure 6-10 Removing Power Harness



PKW0522-97

Removal

1. Shut down the operating system and power down the system.
2. Remove the AC power cords.
3. Expose both the card cage section and the power section of the system (see Section 6.3).
4. Remove the cable clip between the two sections of the system.
5. Unplug the three cable connections to the motherboard and bend the cable back over the power section of the system.
6. Unplug the cable connection to the floppy and, if applicable, to the optional device above the floppy. Bend the cable back over the power section of the system.
7. Unplug the cable connection to the CD-ROM.
8. Unplug the cable connection to the StorageWorks backplane.
9. Remove the power harness from the system.

Replacement

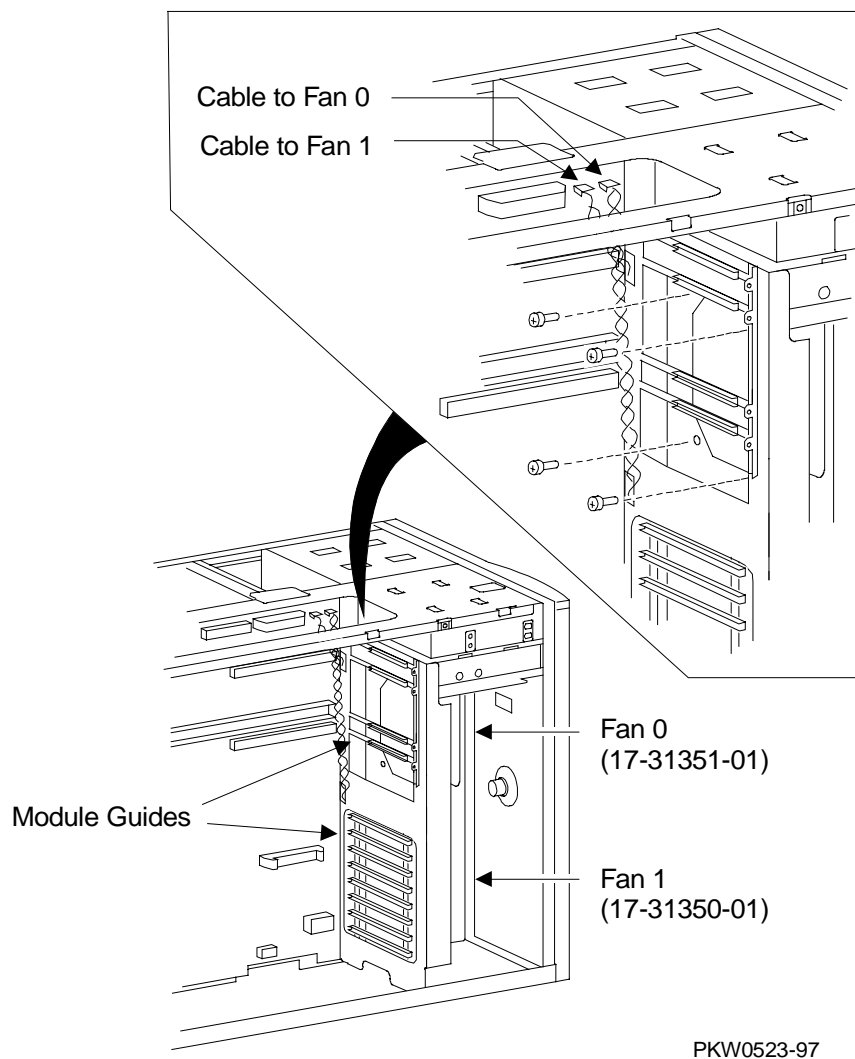
Reverse the steps in the Removal procedure.

Verification

Power up the system.

6.12 System Fan Removal and Replacement

Figure 6-11 Removing System Fan



PKW0523-97

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).

Removing Fan 0

3. Remove the CPU module(s).
4. Remove memory.
5. Trace the wire from the fan to the motherboard to determine which power cord to unplug. Unplug the power cord to fan 0 and pass it through the sheet metal to the fan compartment.
6. Remove the plastic module guides that interfere with access to the four Phillips head screws holding the fan in place.
7. Unscrew the fan from the frame and remove it from the system.

Removing Fan 1

3. Remove any PCI modules that prevent access to the four Phillips head screws that hold fan 1 in place.
4. Remove any plastic module guides that prevent access to the Phillips head screws that hold fan 1 in place.
5. Trace the wire from the fan to the motherboard to determine which power cord to unplug. Unplug the power cord to fan 1 and pass it through the sheet metal to the fan compartment.
6. Unscrew the fan from the frame and remove it from the system.

Replacement

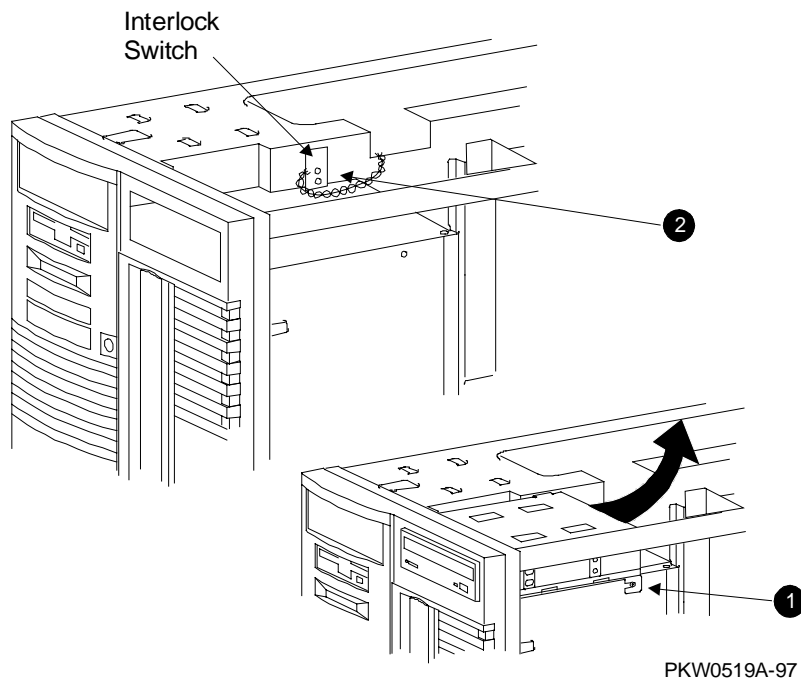
Reverse the steps in the Removal procedure.

Verification

Power up the system. If the fan you installed is faulty, the system will not power up.

6.13 Cover Interlock Removal and Replacement

Figure 6-12 Removing Cover Interlock



Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Loosen the screw that holds the CD-ROM bracket to the system (❶ in Figure 6-12).
4. Detach both the power and the signal connectors at the rear of the CD-ROM.
5. Pull the CD-ROM and the bracket a short distance toward the rear of the system and lift them out of the cabinet.
6. Unplug the interlock switch's pigtail cable from the cable it is connected to.
7. Remove the two screws holding the interlock in place and remove the interlock (❷).

Replacement

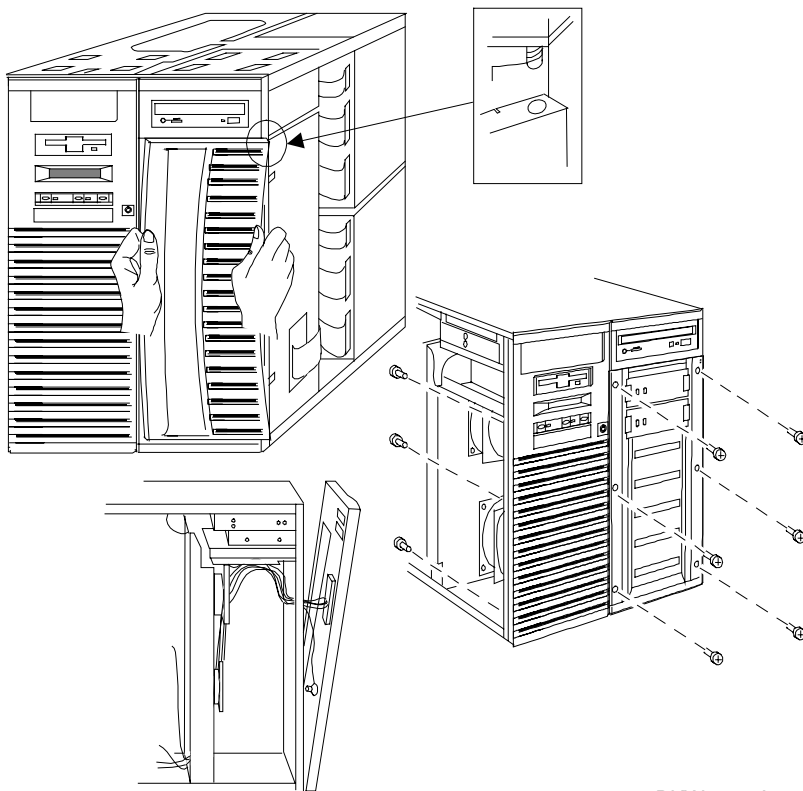
Reverse the steps in the Removal procedure.

Verification

Power up the system. If the switch is faulty, the system will not power up.

6.14 Operator Control Panel Removal and Replacement

Figure 6-13 Removing the OCP



PKW-0501A-97

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. To remove the StorageWorks door:
 - a. Open the door slightly and grab the left edge of the door with your left hand and the right edge of the door with your right hand.
 - b. While pushing the door up, bend it by pulling it away from the system. The door compresses enough so its bottom post slips out of its retaining hole.
 - c. Once the bottom of the door is free, gently pull the top down to release it from the post on the door jam and release it from the spring.
 - d. Put the door aside.
4. Using a Phillips head screwdriver, remove the nine screws holding the molded plastic front panel to the system. (Six screws are accessed from the front of the system and three through the fan compartment of the system.)
5. Tilt the front panel away from the system and disconnect all the cables from the OCP.
6. Once the front panel is removed, unscrew the four screws holding the OCP to the front panel.

Replacement

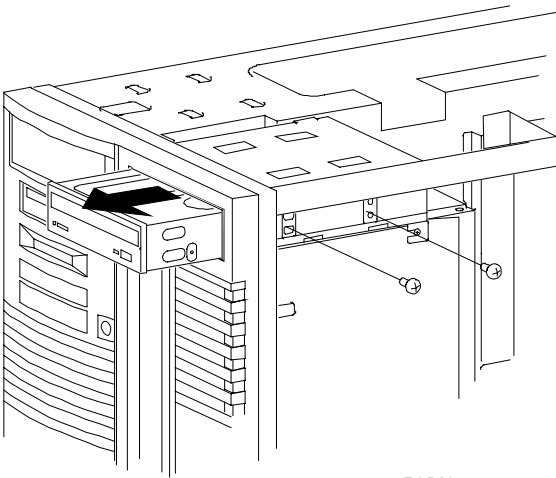
Reverse the steps in the Removal procedure.

Verification

Power up the system. If the OCP you installed is faulty, the system will not power up.

6.15 CD-ROM Removal and Replacement

Figure 6-14 Removing CD-ROM



PKW0519-97

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Loosen the two screws holding the CD-ROM to its bracket (see Figure 6-14).
4. Detach both the power and signal connectors at the rear of the CD-ROM.
5. Pull the CD-ROM forward out of the system.

Replacement

Reverse the steps in the Removal procedure.

Verification

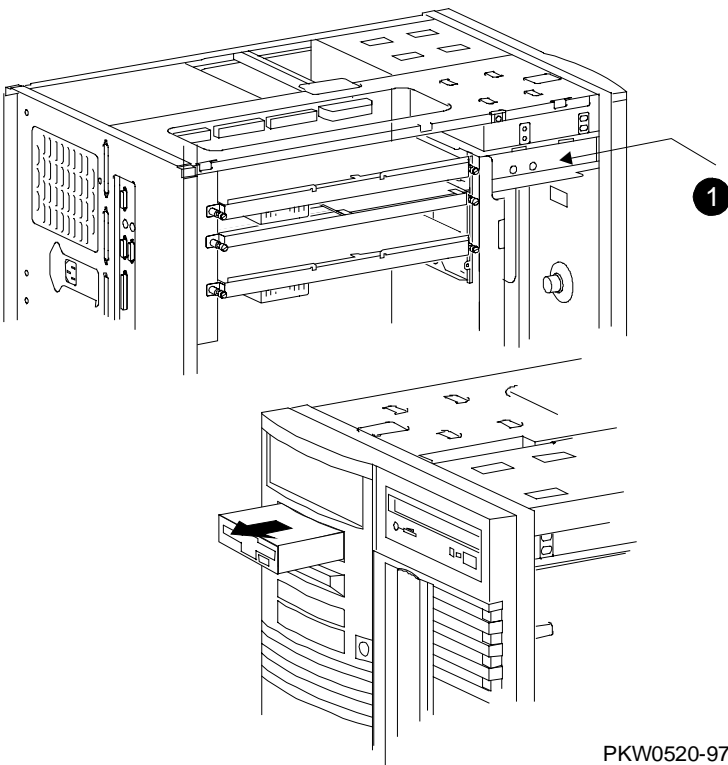
Power up the system. Use the following SRM console commands to test the floppy:

```
P00>>> show dev ncr0  
P00>>> HD buf/dka nnn
```

where *nnn* is the device number; for example, dka500.

6.16 Floppy Removal and Replacement

Figure 6-15 Removing Floppy



PKW0520-97

Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Remove the two Phillips head screws holding the floppy in the system (❶ in Figure 6-15).
4. Slide the floppy out the front of the system.

Replacement

Reverse the steps in the Removal procedure.

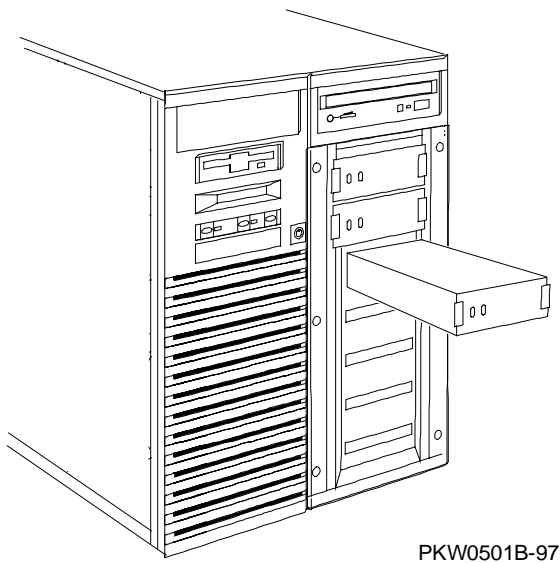
Verification

Power up the system (press the Halt button if necessary to bring up the SRM console). Use the following SRM console commands to test the CD-ROM:

```
P00>>> show dev floppy
P00>>> HD buf/dva0
```

6.17 SCSI Disk Removal and Replacement

Figure 6-16 Removing StorageWorks Disk



Removal

1. Shut down the operating system and power down the system.
2. Open the front door exposing the StorageWorks disks.
3. Pinch the clips on both sides of the disk and slide it out of the shelf.

Replacement

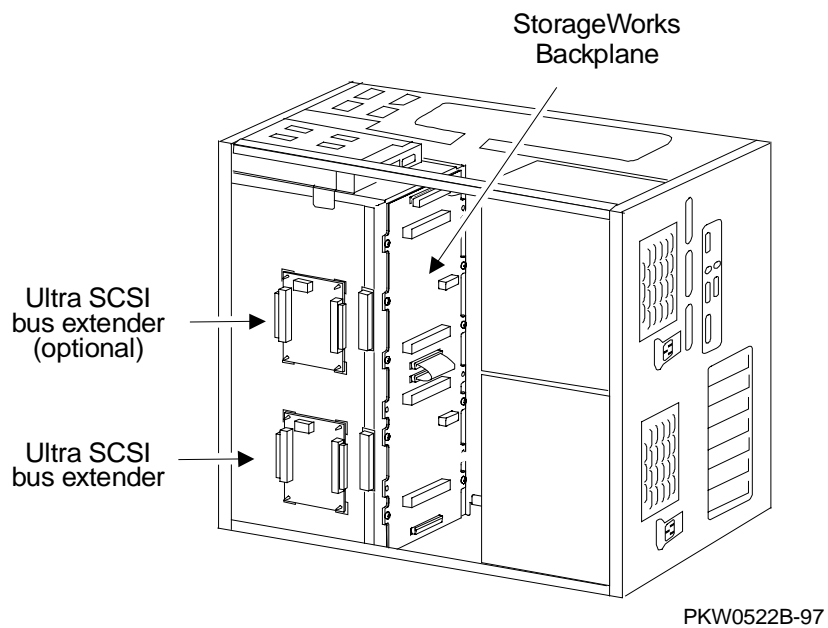
Reverse the steps in the Removal procedure.

Verification

Power up the system. Use the **show device** console commands to verify that the system sees the disk you replaced.

6.18 StorageWorks Backplane Removal and Replacement

Figure 6-17 Removing StorageWorks Backplane



Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system (see Section 6.3).
3. Remove the power and signal cables from the Ultra SCSI bus extender on the side of the StorageWorks shelf.
4. Remove the power harness and all signal cables from the StorageWorks backplane.
5. Using a short Phillips head screwdriver, remove the screws holding the backplane to the back of the shelf and remove from the system.

Replacement

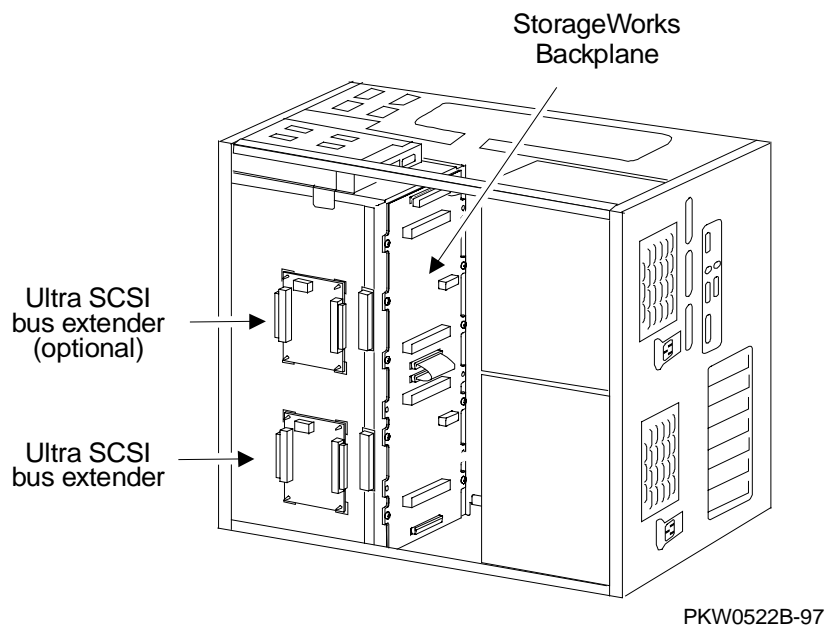
Reverse the steps in the Removal procedure.

Verification

Power up the system. Use the **show device** console command to verify that the StorageWorks shelf is configured into the system.

6.19 StorageWorks Ultra SCSI Bus Extender Removal and Replacement

Figure 6-18 Removing StorageWorks Ultra SCSI Bus Extender



Removal

1. Shut down the operating system and power down the system.
2. Expose the card cage side of the system. See Section 6.3.
3. Remove the power and signal cables from the Ultra SCSI bus extender on the side of the StorageWorks shelf.
4. On early systems the Ultra SCSI bus extender is stuck to the side of the StorageWorks enclosure with adhesive standoffs; in later systems it is mounted on plastic standoffs to which it snaps. If the system has the adhesive, simply pry each corner of the extender free and remove it. If the system has plastic mounts, pinch each with a pair of pliers, free the corner, and pull the bus extender from the enclosure.

Replacement

Reverse the steps in the Removal procedure.

Verification

Power up the system. Use the **show device** console command to verify that the StorageWorks shelf is configured into the system.

Appendix A

Running Utilities

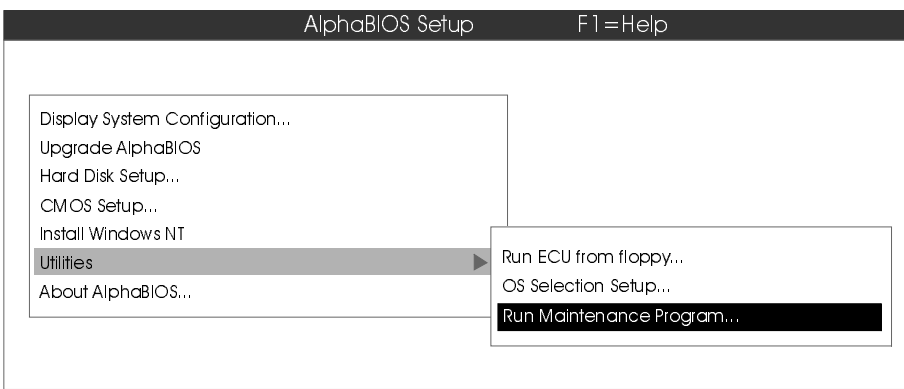
This appendix provides a brief overview of how to load and run utilities. The following topics are covered:

- Running Utilities from a Graphics Monitor
- Running Utilities from a Serial Terminal
- Running ECU
- Running RAID Standalone Configuration Utility
- Updating Firmware with LFU
- Updating Firmware from AlphaBIOS
- Upgrading AlphaBIOS

A.1 Running Utilities from a Graphics Monitor

Start AlphaBIOS and select **Utilities** from the menu. The next selection depends on the utility to be run. For example, to run ECU, select **Run ECU from floppy**. To run RCU, select **Run Maintenance Program**.

Figure A-1 Running a Utility from a Graphics Monitor



PK-0729-96

A.2 Running Utilities from a Serial Terminal

Utilities are run from a serial terminal in the same way as from a graphics monitor. The menus are the same, but some keys are different.

Table A-1 AlphaBIOS Option Key Mapping

AlphaBIOS Key	VTxxx Key
F1	Ctrl/A
F2	Ctrl/B
F3	Ctrl/C
F4	Ctrl/D
F5	Ctrl/E
F6	Ctrl/F
F7	Ctrl/P
F8	Ctrl/R
F9	Ctrl/T
F10	Ctrl/U
Insert	Ctrl/V
Delete	Ctrl/W
Backspace	Ctrl/H
Escape	Ctrl/[

A.3 Running ECU

The EISA Configuration Utility (ECU) is used to configure EISA options on these systems. The ECU can be run either from a graphics monitor or a serial terminal.

1. Start AlphaBIOS Setup. If the system is in the SRM console, issue the command **alphabios**. (If the system has a graphics monitor, you can set the SRM **console** environment variable to **graphics**.)
2. From AlphaBIOS Setup, select Utilities, then select Run ECU from floppy... from the submenu that displays, and press Enter.

NOTE: The EISA Configuration Utility is supplied on diskettes shipped with the system. There is a diskette for Microsoft Windows NT and a diskette for DIGITAL UNIX and OpenVMS.

3. Insert the correct ECU diskette for the operating system and press Enter to run it.

The ECU main menu displays the following options:

EISA Configuration Utility

Steps in configuring your computer

STEP 1: Important EISA configuration information

STEP 2: Add or remove boards

STEP 3: View or edit details

STEP 4: Examine required details

STEP 5: Save and exit

NOTE: Step 1 of the ECU provides online help. It is recommended that you select this step and become familiar with the utility before proceeding.

A.4 Running RAID Standalone Configuration Utility

The RAID Standalone Configuration Utility is used to set up RAID disk drives and logical units. The Standalone Utility is run from the AlphaBIOS Utility menu.

These systems support the KZPSC-xx PCI RAID controller (SWXCR). The KZPSC-xx kit includes the controller, RAID Array 230 Subsystems software, and documentation.

1. Start AlphaBIOS Setup. If the system is in the SRM console, issue the command **alphabios**. (If the system has a graphics monitor, you can set the SRM **console** environment variable to **graphics**.)
2. At the Utilities screen, select Run Maintenance Program. Press Enter.
3. In the Run Maintenance Program dialog box, type **swxcrmgr** in the Program Name: field.
4. Press Enter to execute the program. The Main menu displays the following options:

```
[01.View/Update Configuration]
02.Automatic Configuration
03.New Configuration
04.Initialize Logical Drive
05.Parity Check
06.Rebuild
07.Tools
08.Select SWXCR
09.Controller Setup
10.Diagnostics
```

Refer to the RAID Array Subsystems documentation for information on using the Standalone Configuration Utility to set up RAID drives.

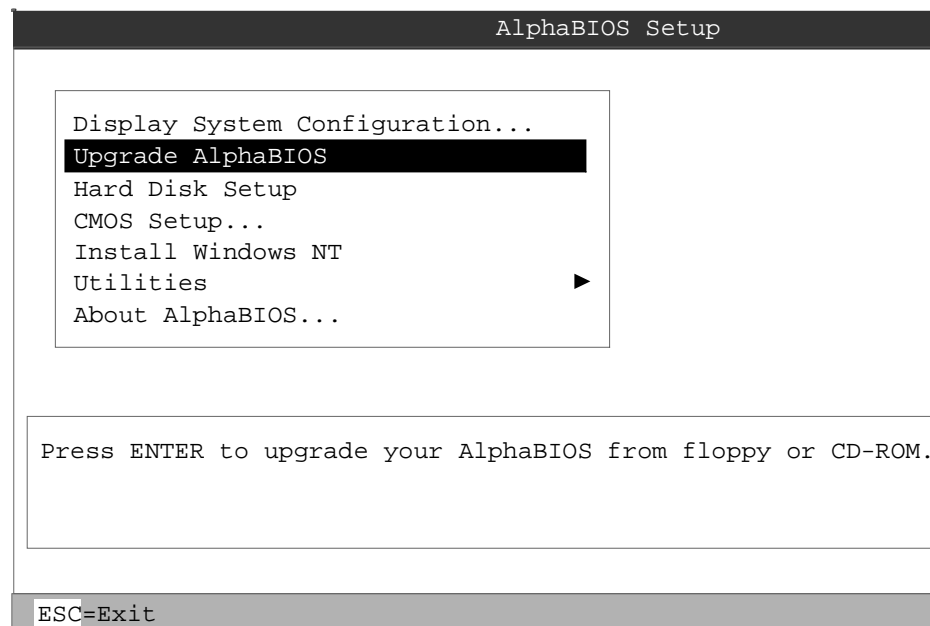
A.5 Updating Firmware with LFU

Start the Loadable Firmware Update (LFU) utility by issuing the `lfu` command at the SRM console prompt or by selecting Update AlphaBIOS in the AlphaBIOS Setup screen. LFU is part of the SRM console.

Example A-1 Starting LFU from the SRM Console

```
P00>>> lfu
***** Loadable Firmware Update Utility *****
Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0
.
UPD>
```

Figure A-2 Starting LFU from the AlphaBIOS Console



PK-0726A-96

Use the Loadable Firmware Update (LFU) utility to update system firmware. You can start LFU from either the SRM console or the AlphaBIOS console.

- From the SRM console, start LFU by issuing the **lfu** command.
- From the AlphaBIOS console, select **Upgrade AlphaBIOS** from the **AlphaBIOS Setup** screen (see Figure A-2).

A typical update procedure is:

1. Start LFU.
2. Use the LFU **list** command to show the revisions of modules that LFU can update and the revisions of update firmware.
3. Use the LFU **update** command to write the new firmware.
4. Use the LFU **exit** command to exit back to the console.

The sections that follow show examples of updating firmware from the local CD-ROM, the local floppy, and a network device. Following the examples is an LFU command reference.

Example A-2 Booting LFU from the CD-ROM

```
P00>>> show dev ncr0
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1      Dka500                      RRD46  1645
P00>>> boot dka500
(boot dka500.5.0.1.1 -flags 0,0)
block 0 of dka500.5.0.1.1 is a valid boot block
.
.
.
jumping to bootstrap code

The default bootfile for this platform is
[AS1200]AS1200_LFU.EXE
Hit <RETURN> at the prompt to use the default bootfile.
```

A.5.1 Updating Firmware from the CD-ROM

Insert the update CD-ROM, start LFU, and select cda0 as the load device.

Example A-3 Updating Firmware from the CD-ROM

***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: cda0

❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS1200FW]: AS1200CP

❷

Copying AS1200CP from DKA500.5.0.1.1 .
Copying [as1200]TCREADME from DKA500.5.0.1.1 .
Copying [as1200]TCSRMRROM from DKA500.5.0.1.1
.....
Copying [as1200]TCARCROM from DKA500.5.0.1.1

```
-----
-
Function      Description
❸
-----
-
Display      Displays the system's configuration table.
Exit         Done exit LFU (reset).
List         Lists the device, revision, firmware name, and
              update revision.
Lfu          Restarts LFU.
Readme       Lists important release information.
Update       Replaces current firmware with loadable data image.
Verify       Compares loadable and hardware images.
? or Help    Scrolls this function table.
-----
-
UPD> list
❹
```

Device	Current Revision	Filename	Update
Revision			
AlphaBIOS	V5.32-0	arcrom	V6.40-1
srmflash	V5.0-1	srmrom	V6.0-3

- ❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, the internal CD-ROM is selected.

- ❷ Select the file that has the firmware update, or press Enter to select the default file. The file options are:

AS1200FW (default) SRM console, AlphaBIOS console, and I/O adapter
firmware

AS1200CP SRM console and AlphaBIOS console firmware only

AS1200IO I/O adapter firmware only

In this example the file for console firmware (AlphaBIOS and SRM) is selected.

- ❸ The LFU function table and prompt (UPD>) display.
- ❹ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the resident firmware for each console (SRM and AlphaBIOS) is at an earlier revision than the firmware in the update file.

Continued on next page

Example A-3 Updating Firmware from the CD-ROM (Continued)

```
UPD> update * 5  
WARNING: updates may take several minutes to complete for each device.  
  
Confirm update on: AlphaBIOS      [Y/(N)] y 6  
  
                        DO NOT ABORT!  
AlphaBIOS      Updating to V6.40-1... Verifying V6.40-1... PASSED.  
  
Confirm update on: srmflash      [Y/(N)] y  
  
                        DO NOT ABORT!  
srmflash      Updating to V6.0-3... Verifying V6.0-3... PASSED.  
  
UPD> exit 7
```

- ⑤ The **update** command updates the device specified or all devices. In this example, the wildcard indicates that all devices supported by the selected update file will be updated.
- ⑥ For each device, you are asked to confirm that you want to update the firmware. The default is no. Once the update begins, do not abort the operation. Doing so will corrupt the firmware on the module.
- ⑦ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

A.5.2 Updating Firmware from the Floppy Disk — Creating the Diskettes

Create the update diskettes before starting LFU. See Section A.4.3 for an example of the update procedure.

Table A-2 File Locations for Creating Update Diskettes on a PC

Console Update Diskette	I/O Update Diskette
AS1200FW.TXT	AS1200IO.TXT
AS1200CP.TXT	TCREADME.SYS
TCREADME.SYS	CIPCA315.SYS
TCSRMMROM.SYS	DFPAA310.SYS
TCARCROM.SYS	KZPAAA11.SYS

To update system firmware from floppy disk, you first must create the firmware update diskettes. You will need to create two diskettes: one for console updates, and one for I/O.

1. Download the update files from the Internet.
2. On a PC, copy files onto two FAT-formatted diskettes.

From an OpenVMS system, copy files onto two ODS2-formatted diskettes as shown in Example A-4.

Example A-4 Creating Update Diskettes on an OpenVMS System

Console Update Diskette

```
$ inquire ignore "Insert blank HD floppy in DVA0, then continue"
$ set verify
$ set proc/priv=all
$ init /density=hd/index=begin dva0: tcods2cp
$ mount dva0: tcods2cp
$ create /directory dva0:[asl200]
$ copy tcreadme.sys dva0:[asl200]tcreadme.sys
$ copy asl200fw.txt dva0:[asl200]asl200fw.txt
$ copy asl200cp.txt dva0:[asl200]asl200cp.txt
$ copy tcsrmrom.sys dva0:[asl200]tcsrmrom.sys
$ copy tcarcrom.sys dva0:[asl200]tcarcrom.sys
$ dismount dva0:
$ set noverify
$ exit
```

I/O Update Diskette

```
$ inquire ignore "Insert blank HD floppy in DVA0, then continue"
$ set verify
$ set proc/priv=all
$ init /density=hd/index=begin dva0: tcods2io
$ mount dva0: tcods2io
$ create /directory dva0:[asl200]
$ create /directory dva0:[options]
$ copy tcreadme.sys dva0:[asl200]tcreadme.sys
$ copy asl200fw.txt dva0:[asl200]asl200fw.txt
$ copy asl200io.txt dva0:[asl200]asl200io.txt
$ copy cipca315.sys dva0:[options]cipca315.sys
$ copy dfpaa310.sys dva0:[options]dfpaa310.sys
$ copy kzpsaA10.sys dva0:[options]kzpsaA10.sys
$ dismount dva0:
$ set noverify
$ exit
```

A.5.3 Updating Firmware from the Floppy Disk — Performing the Update

Insert an update diskette (see Section A.5.2) into the floppy drive. Start LFU and select dva0 as the load device.

Example A-5 Updating Firmware from the Floppy Disk

```
***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0 ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS1200IO,(AS1200CP)]:
AS1200IO ❷

Copying AS1200IO from DVA0 .
Copying TCREADME from DVA0 .
Copying CIPCA315 from DVA0 .
Copying DFPAA252 from DVA0 ...
Copying KZPSAA11 from DVA0 ...

.
. (The function table displays, followed by the UPD> prompt, as
. shown in Example A-3.)

UPD> list ❸
```

Device	Current Revision	Filename	Update Revision
AlphaBIOS	V5.12-3	arcrom	Missing file
pfi0	2.46	dfpaa_fw	2.52
srmflash	T3.2-21	srmrom	Missing file
		cipca_fw	A315
		kzpsa_fw	A11

Continued on next page

- ❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, the internal floppy disk is selected.

- ❷ Select the file that has the firmware update, or press Enter to select the default file. When the internal floppy disk is the load device, the file options are:

AS1200CP (default) SRM console and AlphaBIOS console firmware only

AS1200IO I/O adapter firmware only

The default option in Example A–3 (AS1200FW) is not available, since the file is too large to fit on a 1.44 MB diskette. This means that when a floppy disk is the load device, you can update either console firmware or I/O adapter firmware, but not both in the same LFU session. If you need to update both, after finishing the first update, restart LFU with the **lfu** command and insert the floppy disk with the other file.

In this example the file for I/O adapter firmware is selected.

- ❸ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the update revision for console firmware displays as “Missing file” because only the I/O firmware files are available on the floppy disk.

Continued on next page

Example A-5 Updating Firmware from the Floppy Disk (Continued)

```
UPD> update pfi0 ④  
WARNING: updates may take several minutes to complete for each device.  
  
Confirm update on: pfi0 [Y/(N)] y ⑤  
  
DO NOT ABORT!  
pfi0 Updating to 3.10... Verifying to 3.10... PASSED.  
  
UPD> lfu ⑥  
  
***** Loadable Firmware Update Utility *****  
  
Select firmware load device (cda0, dva0, ewa0), or  
Press <return> to bypass loading and proceed to LFU: dva0  
  
Please enter the name of the options firmware files list, or  
Press <return> to use the default filename [AS1200IO, (AS1200CP)]: ⑦  
  
.   
. (The function table displays, followed by the UPD> prompt.  
. Console firmware can now be updated.)  
  
UPD> exit ⑧
```

- ④ The **update** command updates the device specified or all devices.
- ⑤ For each device, you are asked to confirm that you want to update the firmware. The default is no. Once the update begins, do not abort the operation. Doing so will corrupt the firmware on the module.
- ⑥ The **lfu** command restarts the utility so that console firmware can be updated. (Another method is shown in Example A–6, where the user specifies the file AS1200FW and is prompted to insert the second diskette.)
- ⑦ The default update file, AS1200CP, is selected. The console firmware can now be updated, using the same procedure as for the I/O firmware.
- ⑧ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

Example A–6 Selecting AS1200FW to Update Firmware from the Floppy Disk

```
P00>>> lfu

***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: dva0

Please enter the name of the firmware files list, or
Press <return> to use the default filename [AS1200IO,(AS1200CP)]: as1200fw

Copying AS1200FW from DVA0 .
Copying TCREADME from DVA0 .
Copying TCSRMR0M from DVA0 .....
Copying TCARCROM from DVA0 .....
Copying CIPCA315 from DVA0
Please insert next floppy containing the firmware,
Press <return> when ready. Or type DONE to abort.
Copying CIPCA315 from DVA0 .
Copying DFPAA310 from DVA0 ...
Copying KZPSAA10 from DVA0 ...
.
.
.
```

A.5.4 Updating Firmware from a Network Device

Copy files to the local MOP server's MOP load area, start LFU, and select ewa0 as the load device.

Example A-7 Updating Firmware from a Network Device

***** Loadable Firmware Update Utility *****

Select firmware load device (cda0, dva0, ewa0), or
Press <return> to bypass loading and proceed to LFU: ewa0 ❶

Please enter the name of the options firmware files list, or
Press <return> to use the default filename [AS1200FW]: ❷

Copying AS1200FW from EWA0 .
Copying TCREADME from EWA0 .
Copying TCSRMRROM from EWA0
Copying TCARCROM from EWA0
Copying CIPCA315 from EWA0 .
Copying DFPA310 from EWA0 ...
Copying KZPSA11 from EWA0 ...

.
[The function table displays, followed by the UPD>
prompt, as shown in Example A-3.]

.
UPD> list ❸

Device	Current Revision	Filename	Update Revision
AlphaBIOS	V5.12-2	arcrom	V6.40-1
kzpsa0	A10	kzpsa_fw	A11
kzpsa1	A10	kzpsa_fw	A11
smflash	V1.0-9	smrom	V6.0-3
		cipca_fw	A315
		dfpaa_fw	2.46

Continued on next page

Before starting LFU, download the update files from the Internet (see Preface). You will need the files with the extension .SYS. Copy these files to your local MOP server's MOP load area.

- ❶ Select the device from which firmware will be loaded. The choices are the internal CD-ROM, the internal floppy disk, or a network device. In this example, a network device is selected.
- ❷ Select the file that has the firmware update, or press Enter to select the default file. The file options are:

AS1200FW (default) SRM console, AlphaBIOS console, and I/O adapter firmware

AS1200CP SRM console and AlphaBIOS console firmware only

AS1200IO I/O adapter firmware only

In this example the default file, which has both console firmware (AlphaBIOS and SRM) and I/O adapter firmware, is selected.

- ❸ Use the LFU **list** command to determine the revision of firmware in a device and the most recent revision of that firmware available in the selected file. In this example, the resident firmware for each console (SRM and AlphaBIOS) and I/O adapter is at an earlier revision than the firmware in the update file.

Continued on next page

Example A-7 Updating Firmware from a Network Device (Continued)

```
UPD> update * -all ❹  
WARNING: updates may take several minutes to complete for each  
device.  
  
                DO NOT ABORT!  
AlphaBIOS      Updating to V6.40-1... Verifying V6.40-1...  PASSED.  
  
                DO NOT ABORT!  
kzpsa0         Updating to All    ... Verifying All...  PASSED.  
  
                DO NOT ABORT!  
kzpsa1         Updating to All    ... Verifying All...  PASSED.  
  
                DO NOT ABORT!  
srmflash       Updating to V6.0-3... Verifying V6.0-3...  PASSED.  
  
UPD> exit ❺
```


- ④ The **update** command updates the device specified or all devices. In this example, the wildcard indicates that all devices supported by the selected update file will be updated. Typically, LFU requests confirmation before updating each console's or device's firmware. The **-all** option removes the update confirmation requests.
- ⑤ The **exit** command returns you to the console from which you entered LFU (either SRM or AlphaBIOS).

A.5.5 LFU Commands

The commands summarized in Table A-3 are used to update system firmware.

Table A-3 LFU Command Summary

Command	Function
display	Shows the system physical configuration.
exit	Terminates the LFU program.
help	Displays the LFU command list.
lfu	Restarts the LFU program.
list	Displays the inventory of update firmware on the selected device.
readme	Lists release notes for the LFU program.
update	Writes new firmware to the module.
verify	Reads the firmware from the module into memory and compares it with the update firmware.

These commands are described in the following pages.

display

The **display** command shows the system physical configuration. **Display** is equivalent to issuing the SRM console command **show configuration**. Because it shows the slot for each module, **display** can help you identify the location of a device.

exit

The **exit** command terminates the LFU program, causes system initialization and testing, and returns the system to the console from which LFU was called.

help

The **help** (or **?**) command displays the LFU command list, shown below.

Function	Description
Display	Displays the system's configuration table.
Exit	Done exit LFU (reset).
List	Lists the device, revision, firmware name, and update revision.
Lfu	Restarts LFU.
Readme	Lists important release information.
Update	Replaces current firmware with loadable data image.
Verify	Compares loadable and hardware images.
? or Help	Scrolls this function table.

lfu

The **lfu** command restarts the LFU program. This command is used when the update files are on a floppy disk. The files for updating both console firmware and I/O firmware are too large to fit on a 1.44 MB disk, so only one type of firmware can be updated at a time. Restarting LFU enables you to specify another update file.

list

The **list** command displays the inventory of update firmware on the CD-ROM, network, or floppy. Only the devices listed at your terminal are supported for firmware updates.

The **list** command shows three pieces of information for each device:

- Current Revision — The revision of the device's current firmware
- Filename — The name of the file used to update that firmware
- Update revision — The revision of the firmware update image

readme

The **readme** command lists release notes for the LFU program.

update

The **update** command writes new firmware to the module. Then LFU automatically verifies the update by reading the new firmware image from the module into memory and comparing it with the source image.

To update more than one device, you may use a wildcard but not a list. For example, **update k*** updates all devices with names beginning with k, and **update *** updates all devices. When you do not specify a device name, LFU tries to update all devices; it lists the selected devices to update and prompts before devices are updated. (The default is no.) The **-all** option removes the update confirmation requests, enabling the update to proceed without operator intervention.

*CAUTION: Never abort an **update** operation. Aborting corrupts the firmware on the module.*

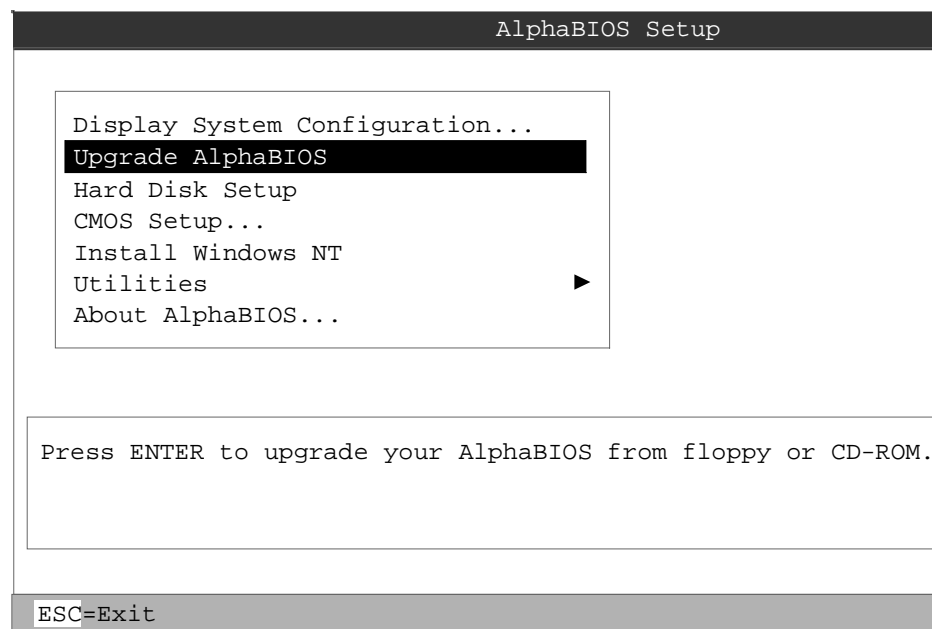
verify

The **verify** command reads the firmware from the module into memory and compares it with the update firmware. If a module already verified successfully when you updated it, but later failed tests, you can use **verify** to tell whether the firmware has become corrupted.

A.6 Updating Firmware from AlphaBIOS

Insert the CD-ROM or diskette with the updated firmware and select Upgrade AlphaBIOS from the main AlphaBIOS Setup screen. Use the Loadable Firmware Update (LFU) utility to perform the update. The LFU exit command causes a system reset.

Figure A-3 AlphaBIOS Setup Screen



PK-0726A-96

A.7 Upgrading AlphaBIOS

It may become necessary to upgrade AlphaBIOS to work with new versions of Windows NT or when enhancements are made.

Use this procedure to upgrade from an earlier version of AlphaBIOS:

1. Insert the diskette or CD-ROM containing the AlphaBIOS upgrade.
2. If you are not already running **AlphaBIOS Setup**, start it by restarting your system and pressing F2 when the **Boot** screen is displayed.
3. In the main **AlphaBIOS Setup** screen, select **Upgrade AlphaBIOS** and press Enter.

The system is reset and the Loadable Firmware Update (LFU) utility is started. See Section A5.5 for LFU commands.

4. When the upgrade is complete, issue the LFU **exit** command. The system is reset and you are returned to AlphaBIOS.

If you press the Reset button instead of issuing the LFU **exit** command, the system is reset and you are returned to LFU.

Appendix B

Halts, Console Commands, and Environment Variables

This appendix discusses halting the system and provides a summary of the SRM console commands and environment variables. The **test** command is described in Chapter 3 of this document. For complete reference information on other SRM commands and environment variables, see your system *User's Guide*.

NOTE: It is recommended that you keep a list of the environment variable settings for systems that you service, because you will need to restore certain environment variable settings after swapping modules. Refer to Table B-4 for a convenient worksheet.

B.1 Halt Button Functions

The Halt button causes the system to perform in various ways depending upon the system state at the time the button is pressed.

When the halt button is pressed, results differ depending upon the state of the machine. Table B-1 describes the full function of the halt button.

Table B-1 Results of Pressing the Halt Button

Machine State	Result
OpenVMS running/hung	SRM console runs
DIGITAL UNIX running/hung	SRM console runs
Windows NT running/hung	Nothing
AlphaBIOS running/hung	Nothing
SRM console running	Sets halt assertion flag; the SRM console continues to run
SRM (1 st 2 secs. of pwr-up)	Nothing
XSROM power-up	Sets halt assertion flag, auto boot ignored
SRM console power-up	Sets halt assertion flag, auto boot ignored

A simple halt causes suspension of a system that is hung or running DIGITAL UNIX or OpenVMS and starts the SRM console.

The halt assertion flag is set in the TOY NVRAM; it is read and cleared by the console only during power-up or reset. When the SRM console finds the halt assertion flag set, the conditions of the environment variables **auto_action = boot/restart** and **os_type = NT** are ignored; the SRM console runs and prints the following message:

```
Halt assertion detected
NVRAM power-up script not executed
AUTO_ACTION=BOOT/RESTART and OS_TYPE=NT ignored, if applicable
P00>>>
```


B.2 Using the Halt Button

Use the Halt button to halt the **DIGITAL UNIX** or **OpenVMS** operating system when it hangs or you want to use the SRM console. Use the Halt button to force Windows NT systems to bring up the SRM console rather than booting or halting in AlphaBIOS.

Using Halt to Shut Down the Operating System

You can use the Halt button if the **DIGITAL UNIX** or **OpenVMS** operating system hangs. Pressing the Halt button halts the operating system back to the SRM console firmware. From the console, you can use the **crash** command to force a crash dump at the operating system level.

The Windows NT operating system does not support halts on this system. Pressing the Halt button during a Windows NT session has no effect.

Using Halt to Clear the Console Password

The SRM console firmware allows you to set a password to prevent unauthorized access to the console. If you forget the password, the Halt button, with the **login** command, lets you clear the password and regain control of the console. See Section 4.8 of your system *User's Guide*.

B.3 Halt Assertion

A halt assertion allows you to disable automatic boots of the operating system so that you can perform tasks from the SRM console.

Under certain conditions, you might want to force a “halt assertion.” A halt assertion differs from a simple halt in that the SRM console “remembers” the halt. The next time you power up, the system ignores the SRM power-up script (nvram) and ignores any environment variables that you have set to cause an automatic boot of the operating system. The SRM console displays this message:

```
Halt assertion detected
NVRAM power-up script not executed
AUTO_ACTION=BOOT/RESTART and OS_TYPE=NT ignored, if applicable
```

Halt assertion is useful for disabling automatic boots of the operating system when you want to perform tasks from the SRM console. It is also useful for disabling the SRM power-up script if you have accidentally inserted a command in the script that will cause a system problem. These conditions are described in the sections “Disabling Autoboot” and “Disabling the SRM Power-Up Script.”

You can force a halt assertion using the Halt button, the RCM **halt** command, or the RCM **haltin** command. Observe the following guidelines for forcing a halt assertion.

Halt Assertion with Halt Button or RCM Halt Command

Press the Halt button on the local system (or enter the RCM **halt** command from a remote system) while the system is powering up or the SRM console is running. The system halts at the SRM console, and the halt status is saved. The next time the system powers up, the saved halt status is checked.

*NOTE: Wait 5 seconds after the system begins powering up before pressing the Halt button or remotely entering the RCM **halt** command.*

Halt Assertion with RCM Haltin Command

Enter the RCM **haltin** command at any time except during power-up. For example, enter **haltin** during an operating system session or when the AlphaBIOS console is running.

If you enter the RCM **haltin** command during a DIGITAL UNIX or OpenVMS session, the system halts back to the SRM console, and the halt status is saved. The next time the system powers up, the saved halt status is checked.

If you enter the RCM **haltin** command when Windows NT or AlphaBIOS is running, the interrupt is ignored. However, you can enter the RCM **haltin** command followed

by the RCM **reset** command to force a halt assertion. Upon reset, the system powers up to the SRM console, but the SRM console does not load the AlphaBIOS console.

Clearing a Halt Assertion

Clear a halt assertion as follows:

- If the halt assertion was caused by pressing the Halt button or remotely entering the RCM **halt** command, the console uses the halt assertion once, then clears it.
- If the halt assertion was caused by entering the RCM **haltin** command, enter the RCM **haltout** command or cycle power on the local system.

Disabling Autoboot

The system automatically boots the selected operating system at power-up or reset if the following environment variables are set:

- For DIGITAL UNIX and OpenVMS, the SRM environment variables **os_type**, **auto_action**, **bootdef_dev**, **boot_file**, and **boot_osflags**
- For Windows NT, the SRM **os_type** environment variable and the **Auto Start** selection in the AlphaBIOS **Standard CMOS Setup** screen

You might want to prevent the system from autobooting so you can perform tasks from the SRM console. Use one of the methods described previously to force a halt assertion. When the SRM console prompt is displayed, you can enter commands to configure or test the system. Chapter 4 of your system *User's Guide* describes the SRM console commands and environment variables.

Disabling the SRM Power-Up Script

The system has a power-up script (file) named “nvram” that runs every time the system powers up. If you accidentally insert a command in the script that will cause a system problem, disable the script by using one of the methods described previously to force a halt assertion. When the SRM console prompt is displayed, edit the script to delete the offending command. See Section 4.4 of your system *User's Guide* for more information on editing the nvram script.

B.4 Summary of SRM Console Commands

The SRM console commands are used to examine or modify the system state.

Table B-2 Summary of SRM Console Commands

Command	Function
alphabios	Loads and starts the AlphaBIOS console.
boot	Loads and starts the operating system.
clear <i>envvar</i>	Resets an environment variable to its default value.
clear password	Sets the password to 0.
continue	Resumes program execution.
crash	Forces a crash dump at the operating system level.
deposit	Writes data to the specified address.
edit	Invokes the console line editor on a RAM file or on the nvram file (power-up script).
examine	Displays the contents of a memory location, register, or device.
halt	Halts the specified processor. (Same as stop .)
help	Displays information about the specified console command.
info <i>num</i>	Displays various types of information about the system: Info shows a list describing the num qualifier. Info 3 reads the impure area that contains the state of the CPU before it entered PAL mode. Info 5 reads the PAL built logout area that contains the data used by the operating system to create the error entry Info 8 reads the IOD and IOD1 registers.
initialize	Resets the system.
lfu	Runs the Loadable Firmware Update Utility.

Continued on next page

Table B-2 Summary of SRM Console Commands (Continued)

Command	Function
login	Turns off secure mode, enabling access to all SRM console commands during the current session.
man	Displays information about the specified console command.
more	Displays a file one screen at a time.
prcache	Initializes and displays status of the PCI NVRAM.
set <i>envvar</i>	Sets or modifies the value of an environment variable.
set host	Connects to an MSCP DUP server on a DSSI device.
set password	Sets the console password or changes an existing password.
set rcm_dialout	Sets a modem dialout string.
set secure	Enables secure mode without requiring a restart of the console.
show <i>envvar</i>	Displays the state of the specified environment variable.
show config	Displays the configuration at the last system initialization.
show cpu	Displays the state of each processor in the system.
show device	Displays a list of controllers and their devices in the system.
show fru	Displays the serial number and revision level of all options.
show memory	Displays memory module information.
show network	Displays the state of network devices in the system.
show pal	Displays the version of the privileged architecture library code (PALcode).
show power	Displays information about the power supplies, system fans, CPU fans, and temperature.
show rcm_dialout	Displays the modem dialout string.
show version	Displays the version of the console program.
start	Starts a program previously loaded on the processor specified.
stop	Halts the specified processor. (Same as halt .)
test	Runs firmware diagnostics for the system.

B.4.1 Summary of SRM Environment Variables

Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. Environment variables are set or changed with the `set envar` command and returned to their default values with the `clear envar` command. Their values are viewed with the `show envar` command. The SRM environment variables are specific to the SRM console.

Table B-3 Environment Variable Summary

Environment Variable	Function
auto_action	Specifies the console's action at power-up, a failure, or a reset.
bootdef_dev	Specifies the default boot device string.
boot_osflags	Specifies the default operating system boot flags.
com*_baud	Changes the default baud rate of the COM1 or the COM2 serial port.
console	Specifies the device on which power-up output is displayed (serial terminal or graphics monitor).
cpu_enabled	Enables or disables a specific secondary CPU.
ew*0_mode	Specifies the connection type of the default Ethernet controller.
ew*0_protocols	Specifies network protocols for booting over the Ethernet controller.
kbd_hardware_type	Specifies the default console keyboard type.
kzpsa*_host_id	Specifies the default value for the KZPSA host SCSI bus node ID.
language	Specifies the console keyboard layout.

Continued on next page

Table B-3 Environment Variable Summary (Continued)

Environment Variable	Function
memory_test	Specifies the extent to which memory will be tested. For DIGITAL UNIX systems only.
ocp_text	Overrides the default OCP display text with specified text.
os_type	Specifies the operating system and sets the appropriate console interface.
pci_parity	Disables or enables parity checking on the PCI bus.
pk*0_fast	Enables fast SCSI mode.
pk*0_host_id	Specifies the default value for a controller host bus node ID.
pk*0_soft_term	Enables or disables SCSI terminators on systems that use the QLogic ISP1020 SCSI controller.
sys_model_num	Displays the system model number and computes certain information passed to the operating system. Must be restored after a PCI motherboard is replaced.
sys_serial_num	Restores the system serial number. Must be set if the system motherboard is replaced.
sys_type	Displays the system type and computes certain information passed to the operating system. Must be restored after a PCI motherboard is replaced.
tga_sync_green	Specifies the location of the SYNC signal generated by the DIGITAL ZLXp-E PCI graphics accelerator option.
tt_allow_login	Enables or disables login to the SRM console firmware on other console ports.

B.5 Recording Environment Variables

This worksheet lists all environment variables. Copy it and record the settings for each system. Use the `show*` command to list environment variable settings.

Table B-4 Environment Variables Worksheet

Environment Variable	System Name	System Name	System Name
auto_action			
bootdef_dev			
boot_osflags			
com1_baud			
com2_baud			
console			
cpu_enabled			
ew*0_mode			
ew*0_protocols			
kbd_hardware_type			
kzpsa*_host_id			
language			
memory_test			
ocp_text			
os_type			
pci_parity			
pk*0_fast			
pk*0_host_id			

Table B-4 Environment Variables Worksheet (Continued)

Environment Variable	System Name	System Name	System Name
pk*0_soft_term			
sys_model_num			
sys_serial_num			
sys_type			
tga_sync_green			
tt_allow_login			

Appendix C

Managing the System Remotely

This chapter describes how to manage the system from a remote location using the remote console manager (RCM). You can use the RCM from a console terminal at a remote location. You can also use the RCM from the local console terminal.

Sections in this chapter are:

- RCM Overview
- First-Time Setup
- RCM Commands
- Dial-Out Alerts
- Using the RCM Switchpack
- Troubleshooting Guide
- Modem Dialog Details

C.1 RCM Overview

The remote console manager (RCM) monitors and controls the system remotely. The control logic resides on the system board.

The RCM is a separate console from the SRM and AlphaBIOS consoles. The RCM is run from a serial console terminal or terminal emulator. A command interface lets you reset, halt, and power the system on or off, regardless of the state of the operating system or hardware. You can also use RCM to monitor system power and temperature.

You can invoke the RCM either remotely or through the local serial console terminal. Once in RCM command mode, you can enter commands to control and monitor the system. Only one RCM session can be active at a time.

- To connect to the RCM remotely, you dial in through a modem, enter a password, and then type an escape sequence that invokes RCM command mode. You must set up the modem before you can dial in remotely.
- To connect to the RCM locally, you type the escape sequence at the SRM console prompt on the local serial console terminal.

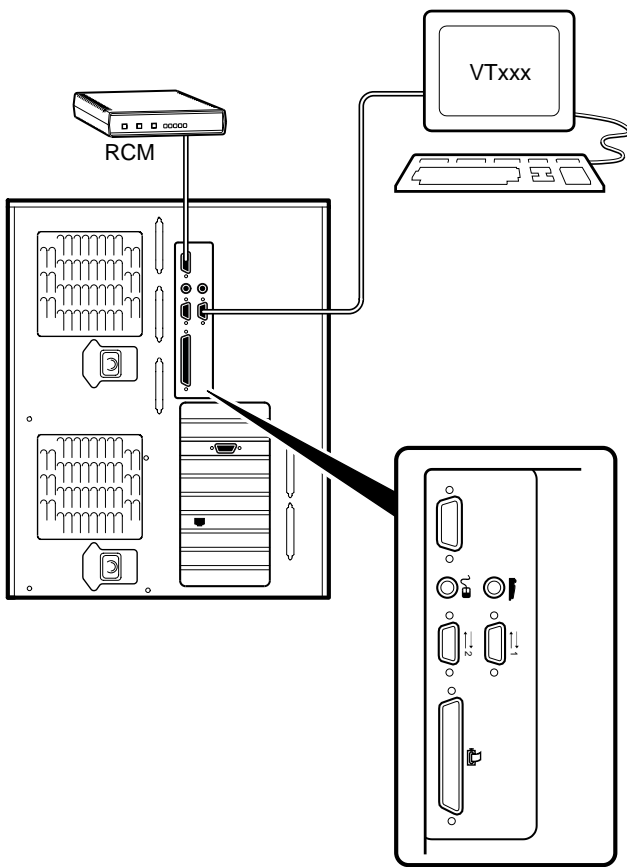
When you are not monitoring the system remotely, you can use the RCM dial-out alert feature. With dial-out alerts enabled, the RCM dials a paging service to alert you about a power failure within the system.

*CAUTION: Do not issue RCM commands until the system has powered up. If you enter certain RCM commands during power-up or reset, the system may hang. In that case you would have to disconnect the power cord at the power outlet. You can, however, use the RCM **halt** command during power-up to force a halt assertion. Refer to Section B.3 for information on halt assertion.*

C.2 First-Time Setup

To set up the RCM to monitor a system remotely, connect the console terminal and modem to the ports at the back of the system, configure the modem port for dial-in, and dial in.

Figure C-1 RCM Connections



PK-0906-97

C.2.1 Configuring the Modem

The RCM requires a Hayes-compatible modem. The controls that the RCM sends to the modem are acceptable to a wide selection of modems. After selecting the modem, connect it and configure it.

Qualified Modems

The modems that have been tested and qualified with this system are:

- Motorola 3400 Lifestyle 28.8
- AT&T Dataport 14.4/FAX
- Hayes Smartmodem Optima 288 V-34/V.FC + FAX

Modem Configuration Procedure

1. Connect a Hayes-compatible modem to the RCM as shown in Figure C-1, and power up the modem.
2. From the local serial console terminal, type the following escape sequence to invoke the RCM:

```
P00>>> ^]^]rcm
```

The character “^” is created by simultaneously holding down the Ctrl key and pressing the] key (right square bracket). The SRM prompt, RCM>, is displayed.

3. Use the **setpass** command to set a modem password.
4. Enable the modem port with the **enable** command.
5. Enter the **quit** command to leave the RCM.
6. You are now ready to dial in remotely.

C.2.2 Dialing In and Invoking RCM

To dial in to the RCM modem port, dial the modem, enter the modem password at the # prompt, and type the escape sequence. Use the hangup command to terminate the session.

A sample dial-in dialog would look similar to the following:

Example C-1 Sample Remote Dial-In Dialog

```
ATQ0V1E1S0=0      ❶
OK
ATDT30167
CONNECT 9600
#                  ❷
RCM V2.0           ❸
RCM>
```

Dialing In and Invoking RCM

1. Dial the number for the modem connected to the modem port. See ❶ in Example C-1 for an example.

The RCM prompts for a password with a “#” character. See ❷.

2. Enter the password that you set with the **setpass** command.

You have three tries to correctly enter the password. After three incorrect tries, the connection is terminated, and the modem is not answered again for 5 minutes. When you successfully enter the password, the RCM banner is displayed. See ❸. You are connected to the system COM1 port, and you have control of the SRM console.

NOTE: At this point no one at the local terminal can perform any tasks except for typing the RCM escape sequence. The local terminal displays any SRM console output entered remotely.

3. Type the RCM escape sequence (not echoed).

```
^]^[rcm
RCM>
```

*NOTE: From RCM command mode, you can change the escape sequence for invoking RCM, if desired. Use the **setesc** command to change the sequence. Be sure to record the new escape sequence.*

4. To terminate the modem connection, enter the RCM **hangup** command.

```
RCM> hangup
```

If the modem connection is terminated without using the **hangup** command or if the line is dropped due to phone-line problems, the RCM will detect carrier loss and initiate an internal **hangup** command. If the modem link is idle for more than 20 minutes, the RCM initiates an auto hangup.

NOTE: Auto hangup can take a minute or more, and the local terminal is locked out until the auto hangup is completed.

C.2.3 Using RCM Locally

Use the default escape sequence to invoke the RCM mode locally for the first time. You can invoke RCM from the SRM console, the operating system, or an application. The RCM quit command reconnects the terminal to the system console port.

1. To invoke the RCM locally, type the RCM escape sequence. See ❶ in Example C-2 for the default sequence.

The escape sequence is not echoed on the terminal or sent to the system. At the RCM> prompt, you can enter RCM commands.

2. To exit RCM and reconnect to the system console port, enter the **quit** command. (see ❷). Press Return to get a prompt from the operating system or system console.

Example C-2 Invoking and Leaving RCM Locally

```
P00>>> ^]^[rcm      ❶  
RCM>
```

```
RCM> quit            ❷  
Focus returned to COM port
```

C.3 RCM Commands

The RCM commands given in Table C-1 are used to control and monitor a system remotely.

Table C-1 RCM Command Summary

Command	Function
alert_clr	Clears alert flag, stopping dial-out alert cycle
alert_dis	Disables the dial-out alert function
alert_ena	Enables the dial-out alert function
disable	Disables remote access to the modem port
enable	Enables remote access to the modem port
halt	Halts the server. Emulates pressing the Halt button and immediately releasing it.
haltin	Causes a halt assertion. Emulates pressing the Halt button and holding it in.
haltout	Terminates a halt assertion created with haltin . Emulates releasing the Halt button after holding it in.
hangup	Terminates the modem connection
help or ?	Displays the list of commands
poweroff	Turns off power. Emulates pressing the On/Off button to the off position.
poweron	Turns on power. Emulates pressing the On/Off button to the on position.
quit	Exits console mode and returns to system console port
reset	Resets the server. Emulates pressing the Reset button.
setesc	Changes the escape sequence for invoking command mode
setpass	Changes the modem access password
status	Displays system status and sensors

Command Conventions

- The commands are not case sensitive.
- A command must be entered in full.
- You can delete an incorrect command with the Backspace key before you press Enter.
- If you type a valid RCM command, followed by extra characters, and press Enter, the RCM accepts the correct command and ignores the extra characters.
- If you type an incorrect command and press Enter, the command fails with the message:

```
*** ERROR - unknown command ***
```

alert_clr

The **alert_clr** command clears an alert condition within the RCM. The alert enable condition remains active, and the RCM will again enter the alert condition if it detects a system power failure.

```
RCM>alert_clr
```

alert_dis

The **alert_dis** command disables RCM dial-out. It also clears any outstanding alerts. Dial-out remains disabled until the **alert_enable** command is issued. See also the **enable** and **disable** commands.

```
RCM>alert_dis
```

alert_ena

The **alert_ena** command enables the RCM to automatically dial out when it detects a power failure within the system. The RCM repeats the dial-out alert at 30-minute intervals until the alert is cleared. Dial-out remains enabled until the **alert_disable** command or the **disable** command is issued. See also the **enable** and **disable** commands.

```
RCM>alert_ena
```

Two conditions must be met for the **alert_enable** command to work:

- A modem dial-out string must be entered from the system console.
- Remote access to the RCM modem port must be enabled with the **enable** command.

If the **alert_enable** command is entered when remote access is disabled, the following message is displayed:

```
*** error ***
```

disable

The **disable** command disables remote access to the RCM modem port. It also disables RCM dial-out.

```
RCM>disable
```

When the modem is disabled, it remains disabled until the **enable** command is issued. If a modem connection is in progress, entering the **disable** command terminates it.

*NOTE: If the modem has been disabled from the RCM switchpack on the motherboard, the **enable** command does not work. To enable the modem, reset the switch 2 (MODEM OFF) on the switchpack to OFF (enabled). See Section C.5 for information on the switchpack.*

enable

The **enable** command enables remote access to the RCM modem port. It can take up to 10 seconds for the **enable** command to be executed.

```
RCM>enable
```

When the modem is enabled, it remains enabled until the **disable** command is issued.

The **enable** command can fail for the following reasons:

- No modem access password was set.
- The initialization string or the answer string might not be set properly. (See Section C.7.)
- The modem is not connected or is not working properly.
- The modem has been disabled from the RCM switchpack. To enable the modem, reset switch 2 (MODEM OFF) on the switchpack to OFF (enabled).

If the **enable** command fails, the following message is displayed:

```
*** ERROR - enable failed ***
```

hangup

The **hangup** command terminates the modem session. When this command is issued, the remote user is disconnected from the server. This command can be issued from either the local or remote console.

```
RCM>hangup
```

halt

The **halt** command halts the managed system. The **halt** command is equivalent to pressing the Halt button on the control panel and then immediately releasing it. The RCM firmware exits command mode and reconnects the user's terminal to the system COM1 serial port.

```
RCM>halt  
Focus returned to COM port
```

The **halt** command can be used to force a halt assertion. See Section B.3 for information on halt assertion.

*NOTE: If you are running Windows NT, the **halt** command has no effect.*

haltin

The **haltin** command halts a managed system and forces a halt assertion. The **haltin** command is equivalent to pressing the Halt button on the control panel and holding it in. This command can be used at any time after system power-up to allow you to perform system management tasks. See Section B.3 for information on halt assertion.

*NOTE: If you are running Windows NT, the **haltin** command does not affect the operating system session, but it does cause a halt assertion.*

haltout

The **haltout** command terminates a halt assertion that was done with the **haltin** command. It is equivalent to releasing the Halt button on the control panel after holding it in (rather than pressing it once and releasing it immediately). This command can be used at any time after system power-up. See Section B.3 for information on halt assertion.

help or ?

The **help** or **?** command displays the RCM firmware commands.

poweroff

The **poweroff** command requests the RCM to power off the system. The **poweroff** command is equivalent to pressing the On/Off button on the control panel to the off position.

```
RCM>poweroff
```

If the system is already powered off or if switch 3 (RPD DIS) on the switchpack has been set to the on setting (disabled), this command has no immediate effect.

To power the system on again after using the **poweroff** command, you must issue the **poweron** command.

If, for some reason, it is not possible to issue the **poweron** command, the local operator can start the system as follows:

1. Press the On/Off button to the off position and disconnect the power cord.
2. Reconnect the power cord and press the On/Off button to the on position.

poweron

The **poweron** command requests the RCM to power on the system. The **poweron** command is equivalent to pressing the On/Off button on the control panel to the on position. For the system power to come on, the following conditions must be met:

- AC power must be present at the power supply inputs.
- The On/Off button must be in the on position.
- All system interlocks must be set correctly.

The RCM exits command mode and reconnects the user's terminal to the system console port.

```
RCM>poweron  
Focus returned to COM port
```

*NOTE: If the system is powered off with the On/Off button, the system will not power up. The RCM will not override the "off" state of the On/Off button. If the system is already powered on, the **poweron** command has no effect.*

quit

The **quit** command exits the user from command mode and reconnects the serial terminal to the system console port. The following message is displayed:

```
Focus returned to COM port
```

The next display depends on what the system was doing when the RCM was invoked. For example, if the RCM was invoked from the SRM console prompt, the console prompt will be displayed when you enter a carriage return. Or, if the RCM was invoked from the operating system prompt, the operating system prompt will be displayed when you enter a carriage return.

reset

The **reset** command requests the RCM to reset the hardware. The **reset** command is equivalent to pressing the Reset button on the control panel.

```
RCM>reset  
Focus returned to COM port
```

The following events occur when the **reset** command is executed:

- The system restarts and the system console firmware reinitializes.
- The console exits RCM command mode and reconnects the serial terminal to the system COM1 serial port.
- The power-up messages are displayed, and then the console prompt is displayed or the operating system boot messages are displayed, depending on how the startup sequence has been defined.

setesc

The **setesc** command resets the default escape sequence for invoking RCM. The escape sequence can be any character string. A typical sequence consists of 2 or more characters, to a maximum of 15 characters. The escape sequence is stored in the module's on-board NVRAM.

NOTE: Be sure to record the new escape sequence. Although the factory defaults can be restored if you forget the escape sequence, this requires resetting the EN RCM switch on the RCM switchpack.

The following sample escape sequence consists of 5 iterations of the Ctrl key and the letter "o".

```
RCM>setesc
^o^o^o^o^o
RCM>
```

If the escape sequence entered exceeds 15 characters, the command fails with the message:

```
*** ERROR ***
```

When changing the default escape sequence, avoid using special characters that are used by the system's terminal emulator or applications.

Control characters are not echoed when entering the escape sequence. Use the **status** command to verify the complete escape sequence.

setpass

The **setpass** command allows the user to change the modem access password that is prompted for at the beginning of a modem session.

```
RCM>setpass
new pass>*****
RCM>
```

The maximum length for the password is 15 characters. If the password exceeds 15 characters, the command fails with the message:

```
*** ERROR ***
```

The minimum password length is one character, followed by a carriage return. If only a carriage return is entered, the command fails with the message:

```
*** ERROR - illegal password ***
```

If you forget the password, you can enter a new password.

status

The **status** command displays the current state of the system sensors, as well as the current escape sequence and alarm information. The following is an example of the display.

```
RCM>status
```

```
Firmware Rev: V2.0
Escape Sequence: ^]^]RCM
Remote Access: ENABLE
Alerts: DISABLE
Alert Pending: NO
Temp (C): 26.0
RCM Power Control: ON
RCM Halt: Deasserted
External Power: ON
Server Power: ON
```

```
RCM>
```

The status fields are explained in Table C-2.

Table C-2 RCM Status Command Fields

Item	Description
Firmware Rev:	Revision of RCM firmware.
Escape Sequence:	Current escape sequence to invoke RCM.
Remote Access:	Modem remote access state. (ENABLE/DISABLE)
Alerts:	Alert dial-out state. (ENABLE/DISABLE)
Alert Pending:	Alert condition triggered. (YES/NO)
Temp (C):	Current system temperature in degrees Celsius.
RCM Power Control:	Current state of RCM system power control. (ON/OFF)
RCM Halt:	Asserted indicates that halt has been asserted with the haltin command. Deasserted indicates that halt has been deasserted with the haltout command or by cycling power with the On/Off button on the control panel. The RCM Halt: field does not report halts caused by pressing the Halt button.
External Power:	Current state of power to RCM. Always on.
Server Power:	Indicates whether power to the system is on or off.

C.4 Dial-Out Alerts

When you are not monitoring the system remotely, you can use the RCM dial-out feature to notify you of a power failure within the system.

When a dial-out alert is triggered, the RCM initializes the modem for dial-out, sends the dial-out string, hangs up the modem, and reconfigures the modem for dial-in. The modem must continue to be powered, and the phone line must remain active, for the dial-out alert feature to work. Also, if you are connected to the system remotely, the dial-out feature does not work.

Enabling Dial-Out Alerts

1. Enter the **set rcm_dialout** command, followed by a dial-out alert string, from the SRM console (see ❶ in Example C-3). See the next topic for details on composing the modem dial-out string.
2. Invoke the RCM and enter the **enable** command to enable remote access dial-in. The RCM **status** command should display “Remote Access: Enable.” See ❷.
3. Enter the **alert_ena** command to enable outgoing alerts. See ❸.

Example C-3 Configuring the Modem for Dial-Out Alerts

```
P00>>> set rcm_dialout "ATDTstring#;" ❶

RCM>enable
RCM>status
.
.
Remote Access:  Enable ❷
.
RCM>alert_ena ❸
```

Composing the Dial-Out String

Enter the **set rcm_dialout** command from the SRM console to compose the dial-out string. Use the **show** command to verify the string. See Example C-4.

Example C-4 Typical RCM Dial-Out Command

```
P00>>> set rcm_dialout "ATXDT9,15085553333,,,,,5085553332#;"  
  
P00>>> show rcm_dialout  
rcm_dialout      ATXDT9,15085553333,,,,,5085553332#;
```

The dial-out string has the following requirements:

- The string cannot exceed 47 characters.
- Enclose the entire string following the **set rcm_dialout** command in quotation marks.
- Enter the characters ATDT after the opening quotation marks. Do not mix case.
- Enter the character X after “AT” if the line to be used also carries voice mail.
- The valid characters for the dial-out string are the characters on a phone keypad: 0–9, *, and #. A comma (,) requests that the modem pause for 2 seconds, and a semicolon (;) is required to terminate the string.

The elements of the dial-out string are explained in Table C-3.

Table C-3 Elements of the Dial-Out String

ATXDT	AT = Attention X = Forces the modem to dial “blindly” (not look for a dial tone). Enter X if the dial-out line modifies its dial tone when used for services such as voice mail. D = Dial T = Tone (for touch-tone) , = Pause for 2 seconds
9,	In the example, “9” gets an outside line. Enter the number for an outside line if your system requires it.
15085553333	Dial the paging service.
, , , , , , ,	Pause for 12 seconds for paging service to answer
5085553332#	“Message,” usually a call-back number for the paging service.
;	Return to command mode. Must be entered at end of string.

C.5 Using the RCM Switchpack

The RCM operating mode is controlled by a switchpack on the system board. Use the switches to enable or disable certain RCM functions, if desired.

Figure C-2 Location of RCM Switchpack on System Board

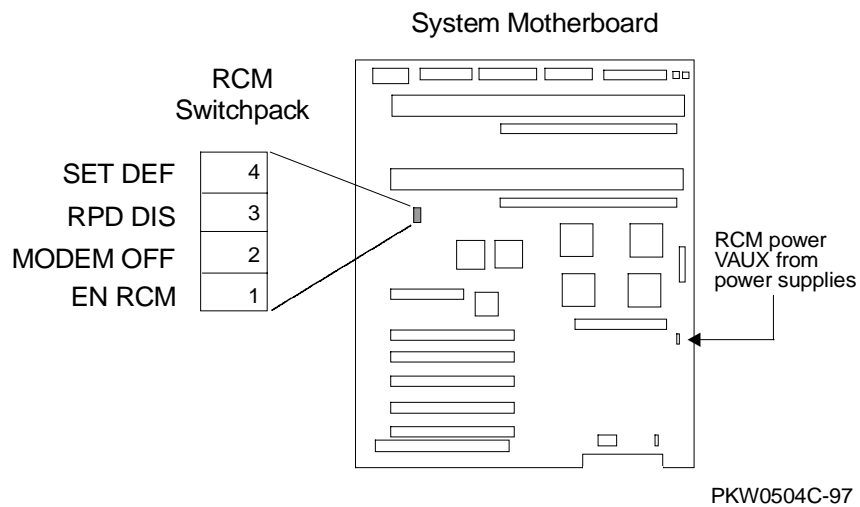
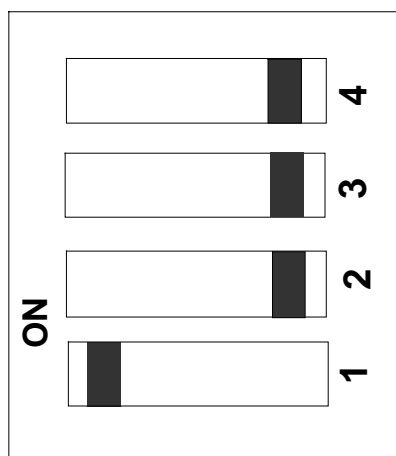


Figure C-3 RCM Switches (Factory Settings)



PKW0950-97

Switch	Name	Description
1	EN RCM	Enables or disables the RCM. The default is ON (RCM enabled). The OFF setting disables RCM.
2	MODEM OFF	Enables or disables the modem. The default is OFF (modem enabled).
3	RPD DIS	Enables or disables remote poweroff. The default is OFF (remote poweroff enabled).
4	SET DEF	Sets the RCM to the factory defaults. The default is OFF (reset to defaults disabled).

Uses of the Switchpack

You can use the RCM switchpack to change the RCM operating mode or disable the RCM altogether. The following are conditions when you might want to change the factory settings.

- Switch 1 (EN RCM)—Set this switch to OFF (disable) if you want to reset the baud rate of the COM1 port to a value other than the system default of 9600. You must disable RCM to select a baud rate other than 9600.
- Switch 2 (MODEM OFF)—Set this switch to ON (disable) if you want to prevent the use of the RCM for monitoring a system remotely. RCM commands can still be run from the local serial console terminal.
- Switch 3 (RPD DIS). Set this switch to ON (disable) if you want to disable the **poweroff** command. With **poweroff** disabled, the monitored system cannot be powered down from the RCM.
- Switch 4 (SET DEF). Set this switch to ON (enable) if you want to reset the RCM to the factory settings. See the section “Resetting the RCM to Factory Defaults.”

Changing a Switch Setting

The RCM switches are numbered on the system board. The default positions are shown in Figure C-3. To change a switch setting:

1. Turn off the system.
2. Unplug the AC power cords.

NOTE: If you do not unplug the power cords, the new setting will not take effect when you power up the system.

3. Remove the system covers. See Section 6.3.
4. Locate the RCM switchpack on the system board and change the switch setting as desired.
5. Replace the system covers and plug in the power cords.
6. Power up the system to the SRM console prompt and type the escape sequence to enter RCM command mode, if desired.

Resetting the RCM to Factory Defaults

You can reset the RCM to factory settings, if desired. You would need to do this if you forgot the escape sequence for the RCM. Follow the steps below.

1. Turn off the system.
2. Unplug the AC power cords.

NOTE: If you do not unplug the power cords, the reset will not take effect when you power up the system.

3. Remove the system covers. See Section 6.3.
4. Locate the RCM switchpack on the system board, and set switch 4 to ON.
5. Replace the system covers and plug in the power cords.
6. Power up the system to the SRM console prompt.

Powering up with switch 4 set to ON resets the escape sequence, password, and modem enable states to the factory defaults.

7. Power down the system, unplug the AC power cords, and remove the system covers.
8. Set switch 4 to OFF.
9. Replace the system covers and plug in the power cords.
10. Power up the system to the SRM console prompt, and type the default escape sequence to invoke RCM command mode:

`^] ^] RCM`

11. Reset the modem password. Reset the escape sequence, if desired, as well as any other states.

C.6 Troubleshooting Guide

Table C-4 is a list of possible causes and suggested solutions for symptoms you might see.

Table C-4 RCM Troubleshooting

Symptom	Possible Cause	Suggested Solution
The local console terminal is not accepting input.	Cables not correctly installed.	Check external cable installation.
	Switch 1 on switchpack set to disable.	Set switch 1 to ON.
	Modem session was not terminated with the hangup command.	Wait several minutes for the local terminal to become active again.
	A remote RCM session is in progress, so the local console terminal is disabled.	Wait for the remote session to be completed.
The console terminal is displaying garbage.	System and terminal baud rate set incorrectly.	Disable RCM and set the system and terminal baud rates to 9600 baud.

Table C-4 RCM Troubleshooting (continued)

Symptom	Possible Cause	Suggested Solution
RCM does not answer when the modem is called.	Modem cables may be incorrectly installed.	Check modem phone lines and connections.
	RCM remote access is disabled.	Enable remote access.
	RCM does not have a valid modem password set.	Set password and enable remote access.
	Switch setting incorrect.	Set switch 1 to ON; switch 2 to OFF.
	The local terminal is currently attached to the RCM.	Enter quit on the local terminal.
	On power-up, the RCM defers initializing the modem for 30 seconds to allow the modem to complete its internal diagnostics and initialization.	Wait 30 seconds after powering up the system and RCM before attempting to dial in.
After the system and RCM are powered up, the COM port seems to hang briefly.	Modem may have had power cycled since last being initialized or modem is not set up correctly.	Enter enable command from RCM.
	This delay is normal behavior.	Wait a few seconds for the COM port to start working.

Continued on next page

Table C-4 RCM Troubleshooting (continued)

Symptom	Possible Cause	Suggested Solution
RCM installation is complete, but system does not power up.	RCM Power Control: is set to DISABLE.	Invoke RCM and issue the poweron command.
You reset the system to factory defaults, but the factory settings did not take effect.	AC power cords were not removed before you reset switch 4 on the RCM switchpack.	Refer to Section C.5.
The remote user sees a “+++” string on the screen.	The modem is confirming whether the modem has really lost carrier. This occurs when the modem sees an idle time, followed by a “3,” followed by a carriage return, with no subsequent traffic. If the modem is still connected, it will remain so.	This is normal behavior.
The message “unknown command” is displayed when the user enters a carriage return by itself.	The terminal or terminal emulator is including a linefeed character with the carriage return.	Change the terminal or terminal emulator setting so that “new line” is not selected.
Cannot enable modem or modem will not answer.	The modem is not configured correctly to work with the RCM.	Modify the modem initialization and/or answer string as described in Section C.7.
	The modem has been disabled on the RCM switchpack.	Refer to Section C.5.

C.7 Modem Dialog Details

This section is intended to help you reprogram your modem if necessary.

Default Initialization and Answer Strings

The modem initialization and answer command strings set at the factory for the RCM are:

Initialization string: AT&F0EVS0=0S12=50<cr>

Answer string ATXA<cr>

NOTE: All modem commands must be terminated with a <cr> character (0x0d hex).

Modifying Initialization and Answer Strings

The initialization and answer strings are stored in the RCM's NVRAM. They come pre-programmed to support a wide selection of modems. With some modems, however, you may need to modify the initialization string, answer string, or both. The following SRM **set** and **show** commands are provided for this purpose.

To replace the initialization string:

```
P00>>> set rcm_init "new_init_string"
```

To replace the answer string:

```
P00>>> set rcm_answer "new_answer_string"
```

To display all the RCM strings that can be set by the user:

```
P00>>> show rcm*
rcm_answer  ATXA
rcm_dialout
rcm_init    AT&F0EVS0=0S12=50
P00>>>
```

Initialization String Substitutions

The following modems require modified initialization strings.

Modem Model	Initialization String
Motorola 3400 Lifestyle 28.8	at&f0e0v0x0s0=2
AT&T Dataport 14.4/FAX	at&f0e0v0x0s0=2
Hayes Smartmodem Optima 288 V-34/V.FC + FAX	at&f0e0v0x0s0=2

Index

?

? command, RCM, C-11

A

Architecture, block diagram, 1-8, 2-6

alert_clr command, RCM, C-8

alert_dis command, RCM, C-8

alert_ena command, RCM, C-8

Alpha 21164 microprocessor, 1-8

Alpha chip composition, 1-11

AlphaBIOS

console, 1-7

loading, 2-7

upgrading, A-25

auto_action environment variable,

SRM, 2-23

B

B3007-AA CPU module, 1-11

B3007-CA CPU module, 1-11

B-cache, 2-21, 2-23

C

CAP chip, 1-21

CAP Error Register, 5-11

CAP Error Register Data Pattern, 4-47

CAP_ERR Register, 5-11

CD-ROM

removal and replacement, 6-30

COM1 port, 2-19

Command codes, 4-55

Command summary (SRM), B-2

Console

SRM, 2-23

Console commands

show fru, 3-15

show memory, 3-14

show power, 3-14

test, 3-8

test memory, 3-10

test pci, 3-12

Console device determination, 2-18

Console device options, 2-19

Console device, changing, 2-19

console environment variable, SRM,
2-21, 2-23

Console power-up tests, 2-16

Control panel, 2-2

display, 2-21

Halt assertion, 1-5

Halt button, 1-4, 1-5

messages in display, 2-3

Reset button, 1-5

Controls

Halt assertion, 1-5

Halt button, 1-5

On/Off button, 1-4

Reset button, 1-5

Cover interlock, 1-3, 1-28

overriding, 1-29

removal and replacement, 6-26

CPU module, 1-10

configuration rules, 1-11

fan removal and replacement, 6-10

removal and replacement, 6-8

variants, 1-11

CPU modules, 1-9, 6-3

D

- Data path chip, 1-21
- DECEvent, 4-6
 - report formats, 4-10
- DIAGNOSE command, 4-7
- Diagnostics, test command, 3-6
- DIMMs, 1-12
 - removal and replacement, 6-14
- disable command, RCM, C-9
- display command (LFU), A-21, A-22
- Double error halt, 4-57, 4-58

E

- ECC syndrome bits, 4-54
- ECU, running, A-4
- EL_ADDR Register, 5-6
- EL_STAT Register, 5-2
- enable command, RCM, C-9
- Environment variables, SRM, 1-7
 - auto_action, 2-23
 - console, 2-21, 2-23
 - os_type, 2-23
 - SRM console, B-4
- Error detector placement, 4-2
- Error log events, 4-5
- Error registers, 5-1
- Event files, translating, 4-7
- Events, filtering, 4-8
- exit command (LFU), A-10, A-16, A-20, A-21, A-22
- External Interface Address Register, 5-6
- External Interface Registers
 - loading and locking rules, 5-7
- External Interface Status Register, 5-2

F

- Fail-safe loader, 2-24
- Fan
 - removal and replacement (CPU chip), 6-10
 - removal and replacement (system), 6-24

- Fans, 6-3

- Fatal errors, 4-5

FEPROM

- and XSROM test flow, 2-13
- contents, 2-5
- defined, 2-5

Firmware

- RCM, C-7
- updating, A-6
- updating from AlphaBIOS, A-24
- updating from CD-ROM, A-7
- updating from floppy disk, A-11, A-13
- updating from network device, A-17
- updating, AlphaBIOS selection, A-5
- updating, SRM command, A-5

Floppy

- removal and replacement, 6-32

- FRU list, 6-2

- FRU part numbers, 6-3

G

- Graphics monitor, VGA, 2-19

H

- halt command, RCM, C-10
- haltin command, RCM, C-11
- haltout command, RCM, C-11

Halts

- caused by power problem, 3-4

- hangup command, RCM, C-10

Hard disk, AlphaBIOS

- error conditions, A-26

- Hard errors, categories of, 4-4

- help command (LFU), A-21, A-22

- help command, RCM, C-11

I

- I squared C bus, 1-34

- INFO 3 command, 4-59

- INFO 5 command, 4-61

- INFO 8 command, 4-63
- Initialization and answer strings
 - default, C-26
 - modifying for modem, C-26
 - substitutions, C-27
- Interlock switches, 6-26
- IOD, 2-23
- IOD detected failure
 - PCI error, 4-32
 - system bus error, 4-27
- IOD error interrupts, 4-5
- IOD, defined, 4-2

L

- LEDs
 - troubleshooting with, 3-2
- LFU
 - exit command, A-22
 - starting, A-5, A-6
 - starting the utility, A-5
 - typical update procedure, A-6
 - update command, A-23
 - updating firmware from CD-ROM, A-7
 - updating firmware from floppy disk, A-11, A-13
 - updating firmware from network device, A-17
- lfu command (LFU), A-14, A-16, A-21, A-22
- LFU commands
 - display, A-21, A-22
 - exit, A-10, A-16, A-20, A-21, A-22
 - help, A-21, A-22
 - lfu, A-14, A-16, A-21, A-22
 - list, A-8, A-14, A-16, A-18, A-20, A-21, A-23
 - readme, A-21, A-23
 - summary, A-21
 - update, A-10, A-21, A-23
 - verify, A-21, A-23
- list command, LFU, A-8, A-14, A-18, A-21, A-23

M

- Machine checks in PAL mode, 4-58
- Maintenance bus, 1-34
- Maintenance bus controller, 1-34
- MC Error Information Register 0, 5-8
- MC Error Information Register 1, 5-9
- MC_ERR0 Register, 5-8
- MC_ERR1 Register, 5-9
- MCHK 620 correctable error, 4-44
- MCHK 630 correctable CPU error, 4-41
- MCHK 660 IOD detected failure, 4-27, 4-32
- MCHK 670 CPU and IOD detected failure, 4-16
- MCHK 670 CPU-detected failure, 4-11
- MCHK 670 read dirty failure, 4-21
- MCHK while in PAL, 4-57
- Memory, 1-12
 - addressing, 1-14
 - addressing rules, 1-15
 - DIMM removal and replacement, 6-14
 - DIMMs, 1-15
 - operation, 1-13
 - option
 - configuration rules, 1-13
 - variants, 1-13
 - riser card removal and replacement, 6-12
- Memory DIMMs, 1-12, 6-3
- Memory errors
 - corrected read data error, 4-53
 - read data substitute error, 4-53
- Memory pairs, 1-13
- Memory riser card, 6-3
 - removal and replacement, 6-12
- Memory tests, 2-14, 2-21
- Memory, broken, 4-53
- Modem
 - dial-in procedure, C-5
 - dialog details, C-26
 - using in RCM, C-3

N

Node IDs, 4-56
NVRAM, 2-3, 2-8

O

Operating the system remotely, C-2
Operator control panel, 1-4
 removal and replacement, 6-28
os_type environment variable, SRM,
 2-7, 2-23

P

Page table entry invalid error, 4-52
PALcode, 2-23
PALcode, described, 4-57
PCI Error Status Register 1, 5-14
PCI master abort, 4-52
PCI parity error, 4-52
PCI slot numbering, 1-23
PCI system error, 4-52
PCI/EISA option removal and
 replacement, 6-18
PCI_ERR Register, 5-14
PIO buffer overflow error
 (PIO_OVFL), 4-51
Power circuit, 1-28
 failures, 1-29
Power cords, 6-4
Power error conditions, 1-27
Power faults, 1-33
Power harness removal and
 replacement, 6-22
Power problems
 at power-up, 3-5
Power supply, 1-30
 fault protection, 1-31
 removal and replacement, 6-20
 voltages, 1-31
Power system components, 6-3
Power up/down sequence, 1-33
poweroff command, RCM, C-11
poweron command, RCM, C-12
Power-up

 SROM and XSROM messages
 during, 2-19

Power-up display, 2-20

Power-up sequence, 2-4

Processor

 determining primary, 2-21

Processor correctable error, 4-5

Processor machine checks, 4-5

Q

quit command, RCM, C-12

R

RCM, C-2, C-19

 changing settings on switchpack, C-
 20

 command summary, C-7

 dial-out alerts, C-16

 invoking and leaving command
 mode, C-6

 modem dialog details, C-26

 modem use, C-3

 remote dial-in, C-5

 resetting to factory defaults, C-22

 switchpack, C-19

 switchpack defaults, C-20

 switchpack location, C-19

 troubleshooting, C-23

 typical dialout command, C-17

RCM commands

 ?, C-11

 alert_clr, C-8

 alert_dis, C-8

 alert_ena, C-8

 disable, C-9

 enable, C-9

 halt, C-10

 haltin, C-11

 haltout, C-11

 hangup, C-10

 help, C-11

 poweroff, C-11

 poweron, C-12

- quit, C-12
- reset, C-12
- setesc, C-13
- setpass, C-13
- status, C-14
- readme command (LFU), A-21, A-23
- Registers, 5-1
- Remote console manager. See RCM
- Remote control switch, 1-25
- Remote dial-in, RCM, C-5
- reset command, RCM, C-12

S

- Safety guidelines, 6-1
- SCSI cables, 6-4
- SCSI Disk removal and replacement, 6-34
- SCSI bus extender removal and replacement, 6-38
- Secure mode
 - releasing, 3-7
- Serial ports, 1-23
- Serial terminal, 2-19
- setesc command, RCM, C-13
- setpass command, RCM, C-13
- Soft errors, categories of, 4-4
- SRM console, 1-7, 2-23
- SRM, 2-21
 - defined, 2-4
 - errors, 2-11
 - power-up test flow, 2-8
 - tests, 2-10
- status command, RCM, C-14
- StorageWorks, 1-36
 - backplane removal and replacement, 6-36
 - disk removal and replacement, 6-34
 - SCSI bus extender removal and replacement, 6-38
- System
 - architecture, 1-8
 - fully configured, 1-9

- System bus, 1-9
- System bus address parity error, 4-50
- System bus block diagram, 1-18
- System bus ECC error, 4-48
- System bus nonexistent address error, 4-49
- System bus to PCI bus bridge, 1-9, 1-20
- System bus to PCI/EISA bus bridge, 1-9
- System cabinet, 1-2
- System cables and jumpers, 6-5
- System components, 1-3
- System consoles, 1-6
- System correctable errors, 4-5
- System drawer
 - remote operation, C-2
- System exposure, 6-6
- System FRU locations, 6-2
- System machine checks, 4-5
- System motherboard, 1-16
 - PCI I/O subsystem section, 1-22
 - power control logic section, 1-26
 - remote control logic section, 1-24
 - removal and replacement, 6-16
 - system bus section, 1-18
 - system bus to PCI bus bridge section, 1-20
- System motherboard LEDs, 3-2

T

- Test command
 - for entire system, 3-8
- Test mem command, 3-10
- Test pci command, 3-12
- Troubleshooting
 - failures at power-up, 3-5
 - IOD detected errors, 4-47
 - power problems, 3-4
 - using error logs, 4-2

U

- Ultra SCSI, 1-36

- cables and jumpers, 6-4
- update command (LFU), A-10, A-16,
A-20, A-21, A-23
- Updating firmware
 - AlphaBIOS console, A-24
 - from AlphaBIOS console, A-5
 - from SRM console, A-5
- Utility programs
 - running from graphics monitor, A-2

V

- verify command (LFU), A-21, A-23

X

- XBUS, 1-23

XSROM

- defined, 2-4
- errors, 2-15
- power-up test flow, 2-12
- tests, 2-13